Fault-tolerant control and fault detection for unmanned aerial vehicles

Fault-Tolerant Control and Fault Detection for Unmanned Aerial Vehicles

Diploma Thesis

Measurement and Control Laboratory

Swiss Federal Institute of Technology

Daniel Rupp

Zürich

Winter 2004/2005

Supervisors: G. Ducard, Dr. E. Shafai, Prof. Dr. H. P. Geering

Abstract

The purpose of this diploma thesis is the presentation of some proposals for the fault tolerant control of unmanned aerial vehicles (UAV). In the first part, the model reference adaptive control (MRAC) scheme is utilized to control UAVs susceptible to actuator failures. In the second part, the multiple model adaptive estimation method (MMAE) is used for the detection and isolation of either actuator or sensor failures. In order to widen the class of detectable failures, the MMAE method is upgraded with extended Kalman filters (EKF) and thus becomes the extended multiple model adaptive estimation method (EMMAE). In this context, each EKF is used for state vector estimation on the one hand and for the estimation of a meaningful failure parameter on the other hand.

1 Introduction

Unmanned aerial vehicles (UAV) are a topic of major interest at the Measurement and Control Laboratory of the Swiss Federal Institute of Technology. Great effort has been put into the area of control and navigation of UAVs. For their practical use, however, safety issues need also to be considered. From a control engineer's point of view, actuator and sensor failures are a principal concern, since during a mission they may lead to catastrophic closed-loop instabilities. This thesis deals with some approaches to cope with such failures and hence to improve the reliability of UAVs. The main difficulty is the uncertainty of failures, i.e., it is usually impossible to predict which sensor or actuator may fail during system operation, or when the failures occur, and their type and values.

In Chapter 2 the model reference adaptive control scheme (MRAC) is used for failure tolerant control of UAVs with actuator failures. The failures are assumed to be of the "lock-in-place"-type, i.e., during operation an actuator may get stuck at an unknown position and thus cause severe system performance deterioration. The MRAC controller is designed such that it uses the remaining (redundant) actuator to match the control system output to the output of a (user defined) reference model, even in the presence of actuator failures. For this model matching, the control parameters are adapted without any explicit knowledge of the failure properties. After the derivation of the control and of the adaptation laws the controller is tested on the nonlinear aircraft model described in Chapter 5.

In Chapter 3 the detection and isolation of actuator as well as sensor failures with the multiple model adaptive estimation method (MMAE) is investigated. The MMAE algorithm is composed of a bank of parallel Kalman filters, each matched to a specific hypothesis about the failure status of the system. During system operation the probabilities of all hypotheses are computed online. The estimated state vectors of each Kalman filter are then blended through a probability-weighted average, thus the MMAE algorithm provides a state variable estimation of the (failed) system as well as the information about the most likely hypothesis. The failures under investigation are assumed to be "hard" failures, which means a complete loss of control authority in case of an actuator and zero mean white noise output in case of a sensor. Further it is assumed that a failed actuator has no more influence on the dynamics of the system. After the introduction of some basic facts of the Kalman filtering theory and the derivation of the hypothesis-testing mechanism, the method is tested on the nonlinear aircraft model described in Chapter 5.

The main drawback of the MMAE method is the need for preliminary knowledge of the failure hypotheses. Since every failure needs to be represented by a Kalman filter, the number of predefined hypotheses is limited by the computational power available. It is therefore impractical to use the MMAE method for failures with unknown failure values (e.g. actuator-lock-in-place failures or sensor bias failures). Hence in Chapter 4 the MMAE method is combined with the parameter estimating ability of an extended Kalman filter (EKF). Where necessary, the Kalman filter in the Kalman filter bank is replaced with an EKF; with this step the failure hypothesis of one single filter can cover all combinations of hard failures with a (slowly varying) bias parameter. After the introduction of some basic facts of the extended Kalman filtering theory and the modelling of appropriate actuator and sensor failures, the method is tested on the nonlinear aircraft model described in Chapter 5.

Chapter 5 provides a short introduction into the basic modelling steps of an aircraft. In the first part, a full six-degree-of-freedom model is derived. For test purposes the resulting MIMO model is then reduced to a SISO model of the lateral dynamics. Furthermore, an easy way to add actuator and sensor redundancy to an existing model is shown in this chapter.

2 Model Reference Adaptive Control of Systems with Actuator Failures

2.1 Introduction

In this chapter the model reference adaptive control method (MRAC) is applied on a linear aircraft model with actuator failures. Consider the linear time-invariant plant described by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}, B = [b_1,...,b_m] \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$ are unknown parameter matrices, $x(t) \in \mathbb{R}^n$ and $y(t) \in \mathbb{R}^p$ are the state vector and the output vector, respectively, and $u(t) = [u_1,...,u_m]^T \in \mathbb{R}^m$ is the input vector whose elements, representing the actuators, may fail during system operation. The plant is so constructed that in the presence of up to any m-q ($1 \le q \le m$) actuator failures the remaining actuators can still achieve a desired control objective In this chapter actuator failures are modeled as lock-in-place, i.e., a failed actuator is stuck at an unknown position and hence is affecting the dynamics of the plant (e.g., a rudder of an airplane is stuck at an angle of five degrees). Assume that the j^{th} actuator is stuck at time t_j at an unknown position $\bar{u}_j$

$$u_j(t) = \bar{u}_j, t \ge t_j,\ j \in \{1,2,...,m\} \tag{2.2}$$

The failures can be modeled as (see [GSX-04])

$$u(t) = v(t) + \sigma(\bar{u} - v(t)) \tag{2.3}$$

where $v(t)$ is the controller output vector and

$$\bar{u} = \left[\bar{u}_1, \bar{u}_2, ..., \bar{u}_m\right]^T, \sigma = diag\{\sigma_1, \sigma_2, ..., \sigma_m\} \tag{2.4}$$

with

$$\sigma_j = \begin{cases} 1 & \text{if the } j^{th} \text{ actuator fails} \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

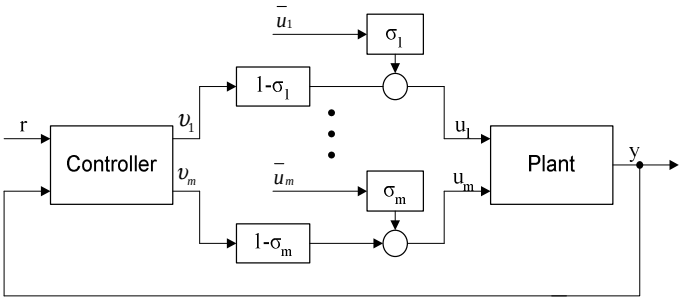

Figure 1: Control system with actuator failures

Assume that the plant with its unknown parameters and the (unknown) failure structure in Figure 1 can be combined into a process with the process-parameter vector θ_P. The MRAC controller shown in Figure 2 consists of a reference model, an adaptation law, and a control law. The goal of the controller is to (asymptotically) match the output signal of the process (with varying parameters θ_P) with the output signal of the (asymptotically) stable reference model by adjusting the parameters of the control law.

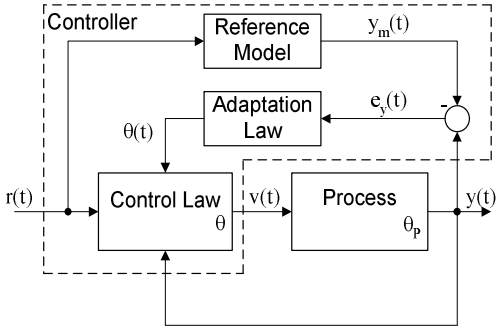

Figure 2: Basic MRAC scheme

2.2 MRAC with Output Feedback

2.2.1 Calculation of Model Reference Matching Parameters

Reconsider the system (2.1). The columns of the matrix B represent the (redundant) actuators of our plant. The actuator group, however, is only driven by a single control signal. This is called an equal actuation scheme (Figure 3)

$$v_1(t) = v_2(t) = ... = v_m(t) \triangleq v_0(t) \tag{2.6}$$

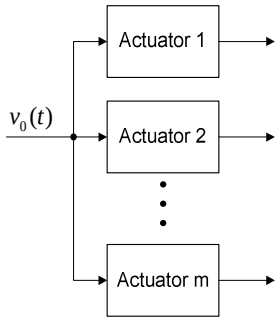

Figure 3: Equal Actuation Scheme

which allows us to treat the actual MIMO system as a (simpler) SISO system. With (2.6) the transfer function of system (2.1) can be written as a sum of all single "actuation channels"

$$G(s) = \sum_{j=1}^{m} \frac{k_{pj} Z_j(s)}{P(s)} \tag{2.7}$$

where k_{pj} is the high-frequency gain (HFG) and $Z_j(s)$ the numerator polynomial of the jth actuator channel. Assuming now that the actuators $j_1,..., j_p$ have failed, this can now be characterized as

$$G(s) = \sum_{j \neq j_1,..., j_p} \frac{k_{pj} Z_j(s)}{P(s)} \triangleq \frac{k_p Z_a(s)}{P(s)} \tag{2.8}$$

In order to design an MRAC controller, all possible failure patterns must satisfy the following conditions:

(C.1) all possible G(s) have the same relative degree n^*

(C.2) all possible G(s) are minimum phase

The first condition implies that the change in the input matrix B due to actuator failures may not change the general structure of the plant, whereas the second condition assures that no unstable zero-pole cancellation takes place. Writing down the reference model in hybrid notation (mixture of time and frequency domains) as

$$y_m(t) = G_m(s)r(t) \tag{2.9}$$

with

$$G_m(s) = \frac{1}{P_m(s)} \tag{2.10}$$

where $P_m(s)$ is a stable monic polynomial (i.e. the highest order coefficient $a_{n^*} = 1$) of degree n^* (which is equal to the previously defined (constant) relative degree of the plant to be controlled). Note that the specific form of (2.10) simplifies the following derivations on the one hand, but restricts the design of the desired reference model on the other hand (e.g., the static gain in general is not equal to one), therefore the real reference signal needs to be prefiltered in order to yield the desired reference output $y_m(t)$. To derive a suitable preliminary filter we write the more general case

$$y_m(t) = \frac{k_m \bar{Z}_m(s)}{\bar{P}_m(s)} \bar{r}(t) \tag{2.11}$$

where $\bar{r}(t)$ is the "real" reference signal and $\bar{P}_m(s)$ is monic stable polynomial. Now (2.11) can be rewritten in the form (2.10) for some $P_m(s)$ and $r(t)$, provided that $P_m(s)\bar{Z}_m(s)$ and $\bar{P}_m(s)$ have the same degree

$$P_m(s)y_m(t) = \frac{k_m P_m(s)\bar{Z}_m(s)}{\bar{P}_m(s)} \bar{r}(t) \triangleq r(t) \tag{2.12}$$

Hence the (desired) reference output $y_m(t)$ can still be achieved with (2.10) by prefiltering (Figure 4) the "real" reference signal $\bar{r}(t)$ with the transfer function

$$G_{pf}(s) = \frac{k_m P_m(s) \bar{Z}_m(s)}{\bar{P}_m(s)} \tag{2.13}$$

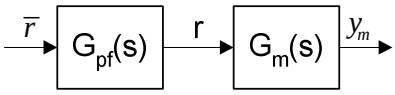

Figure 4: Prefiltering of reference signal

In order to match the output of the plant with the output of the reference model a suitable controller structure needs to be designed. As shown in [SHA-96b], the controller

$$v_0(t) = \theta_1^T \omega_1(t) + \theta_2^T \omega_2(t) + \theta_0 y(t) + k r(t) \tag{2.14}$$

where the auxiliary signals $\omega_1(t)$ and $\omega_2(t)$ defined by

$$\omega_1(t) = \frac{a(s)}{\Lambda(s)} v_0(t), \quad \omega_2(t) = \frac{a(s)}{\Lambda(s)} y(t) \tag{2.15}$$

with $a(s) = \left[1, s, s^2, ..., s^{n-2}\right]$ and $\Lambda(s) = s^{n-1} + \lambda_{n-2} s^{n-2} + ... + \lambda_1 s + \lambda_0$ being a monic stable polynomial of degree n-1 and the parameter vectors $\theta_1 \in \mathbb{R}^{n-1}, \theta_2 \in \mathbb{R}^{n-1}, \theta_0 \in \mathbb{R}, k \in \mathbb{R}$ matches the output of the healthy plant (i.e. no failures) with the reference output if a suitable set of parameters $\theta_1^*, \theta_2^*, \theta_0^*, k^*$ is chosen. The variable k^* represents the ratio of the high frequency gains of the model and the plant.

$$k^* = \frac{1}{k_p} \tag{2.16}$$

However, if at a certain time t, there are p failed actuators, that is $u_j(t) = \bar{u}_j,\ j = j_1, ... j_p,\ 1 \le p \le m$, the plant output may be written as

$$y(t) = G(s) v_0(t) + \bar{y}(t) \tag{2.17}$$

where G(s) is defined in (2.8) and

$$\bar{y}(t) = \sum_{j=j_1,...,j_p} \frac{k_{pj} Z_j(s)}{P(s)} \bar{u}_j(t) \tag{2.18}$$

In order to match the output of the "failed" plant, the control law needs to be extended by a constant term θ_4:

$$v_0(t) = \theta_1^T \omega_1(t) + \theta_2^T \omega_2(t) + \theta_0 y(t) + kr(t) + \theta_4 \tag{2.19}$$

As a next step we want to calculate the matching parameters θ_1^*, θ_2^*, and θ_0^*. With the definitions

$$F_1(s) = \theta_1^{*T} \frac{a(s)}{\Lambda(s)}, \quad F_2(s) = \theta_2^{*T} \frac{a(s)}{\Lambda(s)} \tag{2.20}$$

the matching control signal $v_0(t) = v_0^*(t)$ can be written as

$$\begin{aligned} v_0^*(t) &= F_1(s)v_0^*(t) + F_2(s)G(s)v_0^*(t) + F_2(s)\bar{y}(t) \\ &\quad + \theta_0^* G(s)v_0^*(t) + \theta_0^* \bar{y}(t) + k^* r(t) + \theta_4^* \\ &= \left(F_1(s) + F_2(s)G(s) + \theta_0^* G(s)\right) v_0^*(t) \\ &\quad + F_2(s)\bar{y}(t) + \theta_0^* \bar{y}(t) + k^* r(t) + \theta_4^* \end{aligned} \tag{2.21}$$

which can be solved for $v_0^*(t)$

$$\begin{aligned} v_0^*(t) &= \left(1 - F_1(s) - F_2(s)G(s) - \theta_0^* G(s)\right)^{-1} \\ &\quad \cdot \left(F_2(s)\bar{y}(t) + \theta_0^* \bar{y}(t) + k^* r(t) + \theta_4^*\right) \end{aligned} \tag{2.22}$$

With (2.17) and (2.22), the closed-loop system is

$$\begin{aligned} y(t) &= G(s)\left(1 - F_1(s) - F_2(s)G(s) - \theta_0^* G(s)\right)^{-1} \\ &\quad \cdot \left(F_2(s)\bar{y}(t) + \theta_0^* \bar{y}(t) + k^* r(t) + \theta_4^*\right) + \bar{y}(t) \end{aligned} \tag{2.23}$$

which can be rewritten as

$$\begin{aligned} y(t) &= G(s)\left(1 - F_1(s) - F_2(s)G(s) - \theta_0^* G(s)\right)^{-1} k^* r(t) \\ &\quad + G(s)\left(1 - F_1(s) - F_2(s)G(s) - \theta_0^* G(s)\right)^{-1} \\ &\quad \cdot \left(F_2(s)\bar{y}(t) + \theta_0^* \bar{y}(t) + \theta_4^*\right) + \bar{y}(t) \end{aligned} \tag{2.24}$$

Now if we compare the reference transfer function of (2.9) with the first line of equation (2.24) we obtain

$$G(s)\left(1-F_1(s)-F_2(s)G(s)-\theta_0^*G(s)\right)^{-1}k^*=\frac{1}{P_m(s)} \tag{2.25}$$

and with (2.8)

$$\frac{k_pZ_a(s)}{P(s)}\left(1-F_1(s)-F_2(s)G(s)-\theta_0^*G(s)\right)^{-1}k^*=\frac{1}{P_m(s)} \tag{2.26}$$

Now with $k^*=k_p^{-1}$ from (2.16) we get

$$\frac{Z_a(s)}{P(s)}\left(1-F_1(s)-F_2(s)G(s)-\theta_0^*G(s)\right)^{-1}=\frac{1}{P_m(s)} \tag{2.27}$$

With (2.20) and (2.8) the latter equation can (re)expanded to

$$\frac{Z_a(s)}{P(s)}\left(1-\theta_1^{*T}\frac{a(s)}{\Lambda(s)}-\theta_2^{*T}\frac{a(s)}{\Lambda(s)}\frac{k_pZ_a(s)}{P(s)}-\theta_0^*\frac{k_pZ_a(s)}{P(s)}\right)^{-1}=\frac{1}{P_m(s)} \tag{2.28}$$

After some algebraic operations we get

$$\left(\Lambda(s)-\theta_1^{*T}a(s)\right)P(s)-\left(\theta_2^{*T}a(s)+\theta_0^*\Lambda(s)\right)k_pZ_a(s)=\Lambda(s)Z_a(s)P_m(s) \tag{2.29}$$

Expression (2.29) provides a polynomial equation for the matching parameters $\theta_1^*,\theta_2^*,\theta_0^*$ which can be solved by comparison of coefficients. The existence of a solution for (2.29) is shown in 0. With this equation for the matching parameters we can investigate the convergence properties of the control system with the matching parameters. In particular, we can show that the second and third lines of equation (2.24) go to zero asymptotically. With (2.25) the overall output equation (2.24) can be rewritten as

$$
\begin{aligned}
y(t) &= \frac{1}{P_m(s)} r(t) \\
&\quad + G(s)\left(1 - F_1(s) - F_2(s)G(s) - \theta_0^* G(s)\right)^{-1} \\
&\quad \cdot \left(F_2(s)\bar{y}(t) + \theta_0^* \bar{y}(t) + \theta_4^*\right) + \bar{y}(t) \\
&= \frac{1}{P_m(s)} r(t) \\
&\quad + \frac{1}{k^* P_m(s)}\left(F_2(s)\bar{y}(t) + \theta_0^* \bar{y}(t) + \theta_4^*\right) + \bar{y}(t) \\
&= \frac{1}{P_m(s)} r(t) \\
&\quad + \frac{1}{k^* P_m(s)}\left(\left(F_2(s) + \theta_0^* + k^* P_m(s)\right)\bar{y}(t) + \theta_4^*\right)
\end{aligned}
\tag{2.30}
$$

From (2.25) we have

$$
\left(1 - F_1(s) - F_2(s)G(s) - \theta_0^* G(s)\right) = k^* G(s) P_m(s) \tag{2.31}
$$

and with (2.8)

$$
\left(1 - F_1(s) - F_2(s)\frac{Z_a(s)}{k^* P(s)} - \theta_0^* \frac{Z_a(s)}{k^* P(s)}\right) = \frac{Z_a(s)}{P(s)} P_m(s) \tag{2.32}
$$

which can be rewritten as

$$
F_2(s) + \theta_0^* + k^* P_m(s) = \left(1 - \theta_1^{*T} \frac{a(s)}{\Lambda(s)}\right)\frac{k^* P(s)}{Z_a(s)} \tag{2.33}
$$

Substituting (2.33) into (2.30) yields

$$
\begin{aligned}
y(t) &= \frac{1}{P_m(s)} r(t) \\
&\quad + \frac{1}{k^* P_m(s)}\left(\left(1 - \theta_1^{*T} \frac{a(s)}{\Lambda(s)}\right)\frac{k^* P(s)}{Z_a(s)}\bar{y}(t) + \theta_4^*\right)
\end{aligned}
\tag{2.34}
$$

and with (2.18)

$$
\begin{aligned}
y(t) &= \frac{1}{P_m(s)} r(t) \\
&\quad + \frac{1}{k^* P_m(s)} \left(\left(1 - \theta_1^{*T} \frac{a(s)}{\Lambda(s)} \right) \frac{k^* P(s)}{Z_a(s)} \sum_{j=j_1,\ldots,j_p} \frac{k_{pj} Z_j(s)}{P(s)} \bar{u}_j(t) + \theta_4^* \right) \\
&= \frac{1}{P_m(s)} r(t) \\
&\quad + \frac{1}{k^* P_m(s)} \left(\left(1 - \theta_1^{*T} \frac{a(s)}{\Lambda(s)} \right) \frac{k^*}{Z_a(s)} \sum_{j=j_1,\ldots,j_p} k_{pj} Z_j(s) \bar{u}_j(t) + \theta_4^* \right) \\
&= y_m(t) \\
&\quad + \frac{1}{k^* P_m(s)} \left(\left(1 - \theta_1^{*T} \frac{a(s)}{\Lambda(s)} \right) \frac{k^*}{Z_a(s)} \sum_{j=j_1,\ldots,j_p} k_{pj} Z_j(s) \bar{u}_j(t) + \theta_4^* \right)
\end{aligned}
\tag{2.35}
$$

Note that in the last step the (in general unstable) polynomial P(s) has been cancelled and the remaining polynomials $P_m(s)$, $\Lambda(s)$, $Z_a(s)$ are all stable, therefore a constant θ_4^* exists such that

$$
\lim_{t\to\infty} \frac{1}{k^* P_m(s)} \left(\left(1 - \theta_1^{*T} \frac{a(s)}{\Lambda(s)} \right) \frac{k^*}{Z_a(s)} \sum_{j=j_1,\ldots,j_p} k_{pj} Z_j(s) \bar{u}_j(t) + \theta_4^* \right) = 0 \tag{2.36}
$$

Therefore

$$
\lim_{t\to\infty} (y(t) - y_m(t)) = 0 \tag{2.37}
$$

Note that if no failure has occurred, then from (2.36) we get $\theta_4^* = 0$.

2.2.2 Adaptation Algorithm

In the previous section the matching parameters for a known plant configuration (failure pattern) were derived. However, in general neither the point in time nor the nature of an actuator failure is known a priori and hence an adaptation of the matching parameters is indispensable. In order to derive a stable adaptation law, an important lemma from adaptive control theory [SHA-96b] is needed:

Consider the system in Figure 5

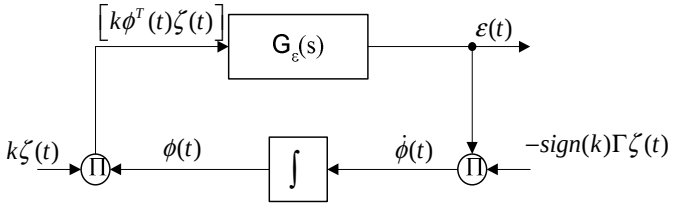

Figure 5: Nonlinear Feedback Structure of Adaptation Law

Lemma 1: (without proof)

If $G_\varepsilon(s)$ is strictly positive real (SPR) ([SHA-96b]) and Γ is a constant positive definite diagonal matrix (adaptation gain) then the (nonlinear) feedback structure in Figure 5 is globally stable and if additionally $\zeta(t)$ is bounded then $\lim_{t\to\infty}\varepsilon(t)=0$ holds.

To derive an error equation, we define

$$\begin{aligned}
\theta^* &= \left[\theta_1^{*T}, \theta_2^{*T}, \theta_0^*, k^*, \theta_4^*\right]^T \in \mathbb{R}^{2n+1} \\
\theta(t) &= \left[\theta_1^T(t), \theta_2^T(t), \theta_0(t), k(t), \theta_4(t)\right]^T \in \mathbb{R}^{2n+1} \\
\omega(t) &= \left[\omega_1^T(t), \omega_2^T(t), y(t), r(t), 1\right]^T \in \mathbb{R}^{2n+1} \\
\phi(t) &= \theta(t) - \theta^*
\end{aligned} \tag{2.38}$$

The control law (2.19) can now be written as

$$\begin{aligned} u(t) &= \left[\phi(t) + \theta^*\right]^T \omega(t) \\ &= \phi^T(t)\omega(t) + \theta^{*T}\omega(t) \end{aligned} \tag{2.39}$$

With (2.39) the structure in Figure 6 can be derived

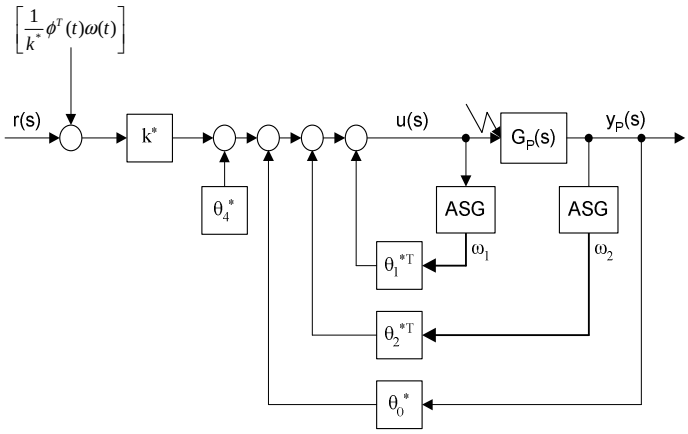

Figure 6: Alternative Representation of the Adaptive Control Law. ASG stands for "auxiliary signal generator".

Since the parameters θ^* are the "true" ones the output of the structure in Figure 6 can be written as

$$y(t) = G_m(s)r(t) + G_m(s)\left[\frac{1}{k^*}\phi^T(t)\omega(t)\right] \tag{2.40}$$

Note that because of the perfect model matching the real plant $G_p(s)$ has been replaced by the reference model $G_m(s)$. Now with

$$y_m(t) = G_m(s)r(t) \tag{2.41}$$

the matching error can be written as

$$e(t) = G_m(s)\left[\frac{1}{k^*}\phi^T(t)\omega(t)\right] \tag{2.42}$$

However, Lemma 1 can not be applied directly since the reference model is not necessarily SPR (the relative order of the plant is in general greater than one). In this case the matching error (2.42) needs to be extended with an additional error signal υ

$$\begin{aligned}\varepsilon(t) &= e(t) + \upsilon(t) \\ &= e(t) + G_m(s)L(s)\frac{1}{k^*}\left[\phi^T(t)L^{-1}(s)I - L^{-1}(s)\phi^T(t)\right]\omega(t)\end{aligned} \tag{2.43}$$

The expression $L^{-1}(s)I$ implies that every signal of the vector $\omega(t)$ is filtered through "its own" transfer function $L^{-1}(s)$ (which is a signal theory issue and not a mathematical one). Now with an appropriate choice of L the transfer function $G_m L$ can be made SPR. In this work the obvious choice

$$L(s) = G_m(s)^{-1} = P_m(s) \tag{2.44}$$

is made such that

$$G_m(s)L(s) = 1 \text{ (SPR)} \tag{2.45}$$

and (2.43) may be written as

$$\varepsilon(t) = e(t) + \frac{1}{k^*}\left[\phi^T(t)G_m(s)I - G_m(s)\phi^T(t)\right]\omega(t) \tag{2.46}$$

Note that $\phi(t)$ is not available in (2.46) but with $\phi(t) = \theta(t) - \theta^*(t)$ we get

$$\begin{aligned}\left[\phi^T(t)G_m(s)I - G_m(s)\phi^T(t)\right]\omega(t) &= \left[\left[\theta(t) - \theta^*\right]^T G_m(s)I - G_m(s)\left[\theta(t) - \theta^*\right]^T\right]\omega(t) \\ &= \left[\theta^T(t)G_m(s)I - G_m(s)\theta^T(t)\right]\omega(t) + \left[G_m(s)\theta^{*T} - \theta^{*T}G_m(s)I\right]\omega(t)\end{aligned} \tag{2.47}$$

and since $\theta^*(t)$ is constant

$$\left[G_m(s)\theta^{*T} - \theta^{*T}G_m(s)I\right]\omega(t) = 0 \tag{2.48}$$

and hence

$$\left[\phi^T(t)G_m(s)I - G_m(s)\phi^T(t)\right]\omega(t) = \left[\theta^T(t)G_m(s)I - G_m(s)\theta^T(t)\right]\omega(t) \tag{2.49}$$

Now the extended error signal can be rewritten as

$$\varepsilon(t) = e(t) + \frac{1}{k^*}\left[\theta^T(t)G_m(s)I - G_m(s)\theta^T(t)\right]\omega(t) \tag{2.50}$$

However, since the parameter k^* in (2.50) is not known, it is replaced by an additional gain $k_1(t)$

$$\varepsilon(t) = e(t) + k_1(t)\left[\theta^T(t)G_m(s)I - G_m(s)\theta^T(t)\right]\omega(t) \tag{2.51}$$

where $k_1(t)$ can be seen as the sum of the "true" parameter k^{*-1} and the corresponding parameter error $\rho(t)$

$$k_1(t) = k^{*-1} + \rho(t) \tag{2.52}$$

Hence (2.50) can be written as

$$\varepsilon(t) = e(t) + \left(k^{*-1} + \rho(t)\right)\left[\theta^T(t)G_m(s)I - G_m(s)\theta^T(t)\right]\omega(t) \tag{2.53}$$

and with (2.42) the expression (2.53) can be rewritten as

$$\begin{aligned}
\varepsilon(t) &= G_m(s)\left[k^{*-1}\phi^T(t)\omega(t)\right] + \left(k^{*-1} + \rho(t)\right)\left[\theta^T(t)G_m(s)I - G_m(s)\theta^T(t)\right]\omega(t) \\
&= G_m(s)\left[k^{*-1}\left(\theta^T(t) - \theta^{*T}\right)\omega(t)\right] + \left(k^{*-1} + \rho(t)\right)\left[\theta^T(t)G_m(s)I - G_m(s)\theta^T(t)\right]\omega(t) \\
&= \cancel{G_m(s)k^{*-1}\theta^T(t)\omega(t)} - G_m(s)k^{*-1}\theta^{*T}\omega(t) \\
&\quad + k^{*-1}\theta^T(t)G_m(s)I\omega(t) - \cancel{k^{*-1}G_m(s)\theta^T(t)\omega(t)} \\
&\quad + \rho(t)\left[\theta^T(t)G_m(s)I - G_m(s)\theta^T(t)\right]\omega(t) \\
&= k^{*-1}\theta^T(t)G_m(s)I\omega(t) - G_m(s)k^{*-1}\theta^{*T}\omega(t) + \rho(t)\left[\theta^T(t)G_m(s)I - G_m(s)\theta^T(t)\right]\omega(t)
\end{aligned} \tag{2.54}$$

Using the result of (2.48) the term $G_M(s)k^{*-1}\theta^{*T}\omega(t)$ can be seen as $k^{*-1}\theta^{*T}(t)G_M I\omega(t)$ and hence the last line of (2.54) writes

$$\varepsilon(t) = k^{*-1}\varphi^T(t)\zeta(t) + \rho(t)e_2(t) \tag{2.55}$$

with

$$\zeta(t) = G_m I\omega(t) \tag{2.56}$$

and

$$e_2(t) = \left[\theta^T(t)G_m I - G_m\theta^T(t)\right]\omega(t) \tag{2.57}$$

Now Lemma 1 can be used to compute the adaptation law for $\phi(t)$ and $\rho(t)$

$$\begin{aligned}\dot{\phi}(t) &= \dot{\theta}(t) = -sign\left(\frac{1}{k^*}\right)\Gamma\zeta(t)\varepsilon(t)\\ \dot{\rho}(t) &= -\gamma e_2(t)\varepsilon(t)\end{aligned} \tag{2.58}$$

Note that $\gamma > 0 \in \mathbb{R}$ in the second adaptation law in (2.58) corresponds to Γ and $e_2(t)$ to $\zeta(t)$. In order to guarantee all closed-loop signals to be bounded and hence to ensure that $\lim_{t\to\infty}\left(y(t) - y_m(t)\right) = 0$ the adaptation law needs to be normalized with $1 + \zeta^T(t)\zeta(t) + e_2^2$ ([NAR-89]).

$$\begin{aligned}\dot{\phi}(t) &= \dot{\theta}(t) = \frac{-sign\left(\frac{1}{k^*}\right)\Gamma\zeta(t)\varepsilon(t)}{1 + \zeta^T(t)\zeta(t) + e_2^2}\\ \dot{\rho}(t) &= \dot{k}_1(t)\frac{-\gamma e_2(t)\varepsilon(t)}{1 + \zeta^T(t)\zeta(t) + e_2^2}\end{aligned} \tag{2.59}$$

2.3 Simulation Results

In this section some simulation results are presented. For the simulation a nonlinear model of the vertical dynamics of a model aircraft is used (see Section 5.3). The nonlinear model is linearized around an equilibrium point

$$x_{nom}(t) = \begin{bmatrix} 0.00907714 \\ 0 \\ 67.7244 \\ 0.614761 \end{bmatrix} \begin{matrix} \text{pitch attitude } [\text{rad}] \\ \text{pitch rate } [\text{rad/s}] \\ \text{longitudinal velocity } [\text{m/s}] \\ \text{normal velocity } [\text{m/s}] \end{matrix} \tag{2.60}$$

In order to apply the failure tolerant control algorithm we add three (redundant) elevators according to Section 5.2

$$u_{red}(t) = \begin{bmatrix} c_{u1}u_1(t) \\ c_{u2}u_2(t) \\ c_{u3}u_3(t) \end{bmatrix} \tag{2.61}$$

where c_{ui} is the efficiency coefficient of the ith elevator. For our simulation we choose $c_{u1} = 1,\ c_{u2} = 0.9,\ c_{u3} = 1.1$. The nominal input values are chosen as

$$u_{1,nom}(t) = 0.00560317,\ u_{2,nom}(t) = 0,\ u_{3,nom}(t) = 0 \tag{2.62}$$

With these preliminaries and the parameter values from Appendix A.2 we get the state space model

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -94.5389 & 0.0126704 & -1.39582 \\ -9.8096 & -0.614761 & -0.151431 & -0.456487 \\ -0.0890455 & 67.7244 & -0.14478 & -15.964 \end{bmatrix},\ B = \begin{bmatrix} 0 & 0 & 0 \\ 153.153 & 137.838 & 168.468 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.63}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

Obviously the three resulting transfer functions are proportional in accordance with to the efficiency coefficients. Hence we get

$$G_i(s) = c_{ui}\frac{153s^2 + 2468s + 360.1}{s^4 + 110.7s^3 + 1620s^2 + 237.2s + 3.947},\ i = 1...3 \tag{2.64}$$

Based on the transfer functions (2.64) the parameters for the adaptive controller are calculated. However, various simulations show that the performance of the adaptive control is only reasonable with very fast reference models. For the following simulations the following (tuning) parameters are used

$$\begin{aligned} &G_m(s) = \frac{1}{s+50} \cdot \frac{1}{s+50},\ \Lambda(s) = (s+10)(s+10)(s+10) \\ &\Gamma = diag\{0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}, \\ &\gamma = 1 \end{aligned} \tag{2.65}$$

With these parameters the model matching controller parameters (for the no-failure case) can be calculated. The matching parameter are calculated with Mathematica.

$$\begin{aligned} &\theta_1^* = \begin{bmatrix} 954.512 \\ -14.115 \\ -5.4611 \end{bmatrix},\ \theta_2^* = \begin{bmatrix} -1003.42 \\ -508.725 \\ -28.8334 \end{bmatrix}, \\ &\theta_0^* = -4.43759,\ k^* = 0.00217647,\ \theta_4^* = 0 \end{aligned} \tag{2.66}$$

Figure 7 shows the result for a step of the pitch attitude with and without adaptation of the controller parameters. Since the matching parameters are chosen as initial values, the plant

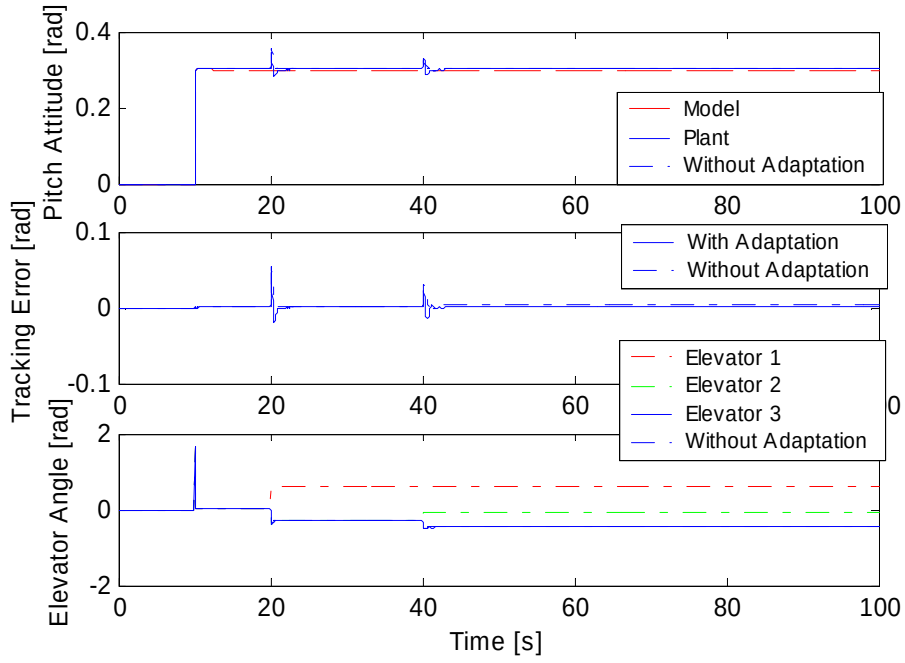

Figure 7: Simulated step of pitch attitude (with linear model). Elevator 1 fails after 20s and Elevator 2 after 40s.

matches the model perfectly during the first 20s. At 20s elevator 1 and at 40s elevator 2 fails and the controller compensates for the respective actuator bias. If we compare the case with and without adaptation there is only a small difference. This is due to the high controller gains shown in (2.66) which keep the tracking error small, even in case of no adaptation. Figure 8 shows the changes of the controller parameters; all parameters show nearly no changes.

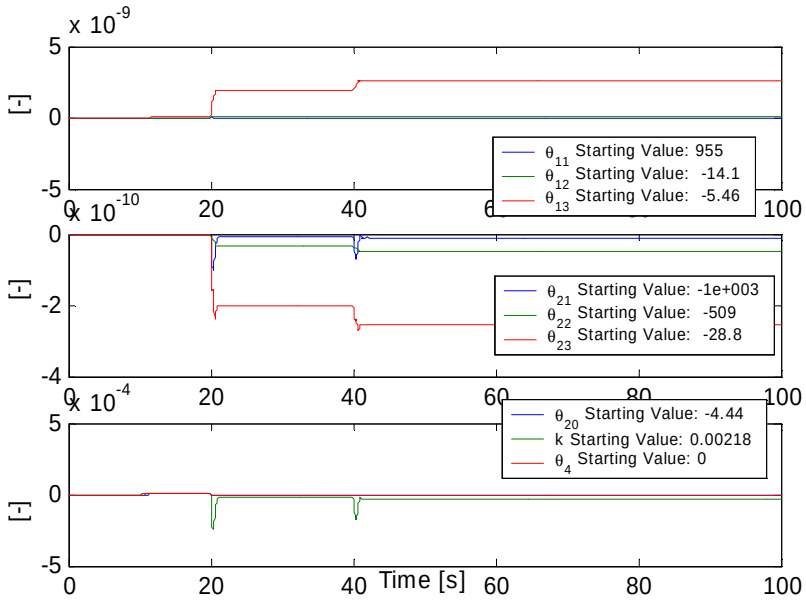

Figure 8: Changes of the controller parameters. All parameters are hardly changed.

The controller developed above is not feasible for real application due to the high parameter gains mentioned. Moreover, further simulations show that any controller designed based on the transfer function (2.64), tends to be very sensitive towards changes of the tuning parameters; if the reference model is chosen too slow the control system even diverges. Furthermore, if we look at the Bode diagram of the plant and at its poles (2.67), we see that the bandwidth of 50 rad/s of the reference model is chosen far too high compared to the bandwidth of the system at 0.0191 rad/s.

$$s_1 = -0.0191\,rad/s,\ s_2 = -0.1286\,rad/s,\ s_3 = -17.1898\,rad/s,\ s_4\ -93.3168\,rad/s \qquad (2.67)$$

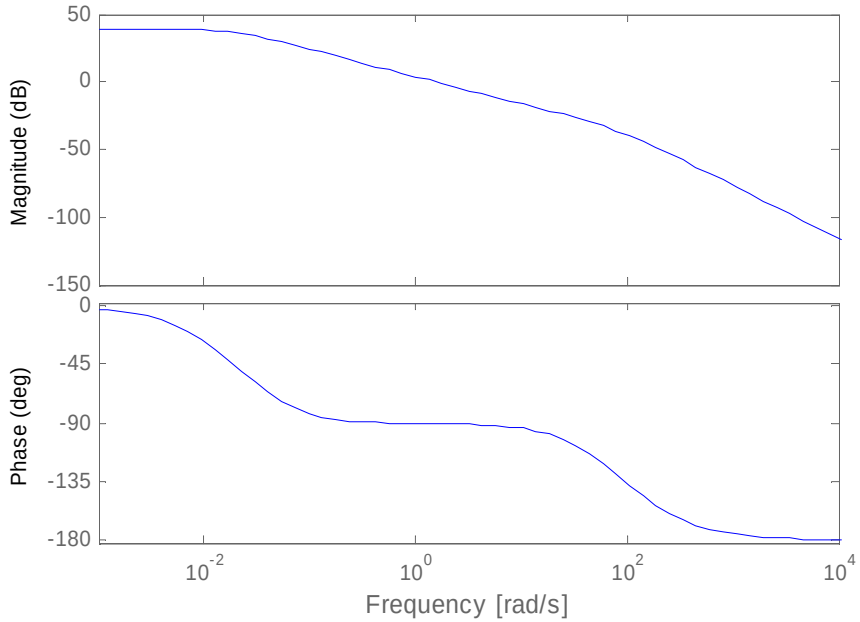

Figure 9: Bode plot of the transfer function $G_1(s)$ from (2.64).

In order to solve this problem the order of the plant is reduced [GEE-99], i.e. unimportant dynamics are omitted. The joint grammians of the balanced system are

$$\bar{W}_c = \bar{W}_o = diag\{80.2546, 0.9739, 0.0125, 0.0041\} \tag{2.68}$$

This implies that the last two state variables of the balanced system can be omitted, which leads to the new transfer functions

$$G_{red,i}(s) = c_{ui}\frac{1.519s + 0.2323}{s^2 + 0.1523s + 0.002546},\ i = 1...3 \tag{2.69}$$

If we compare the Bode diagrams of (2.64) and (2.69) (see Figure 10) we see that down to the magnitude of about -30 dB the reduced transfer function is a good approximation.

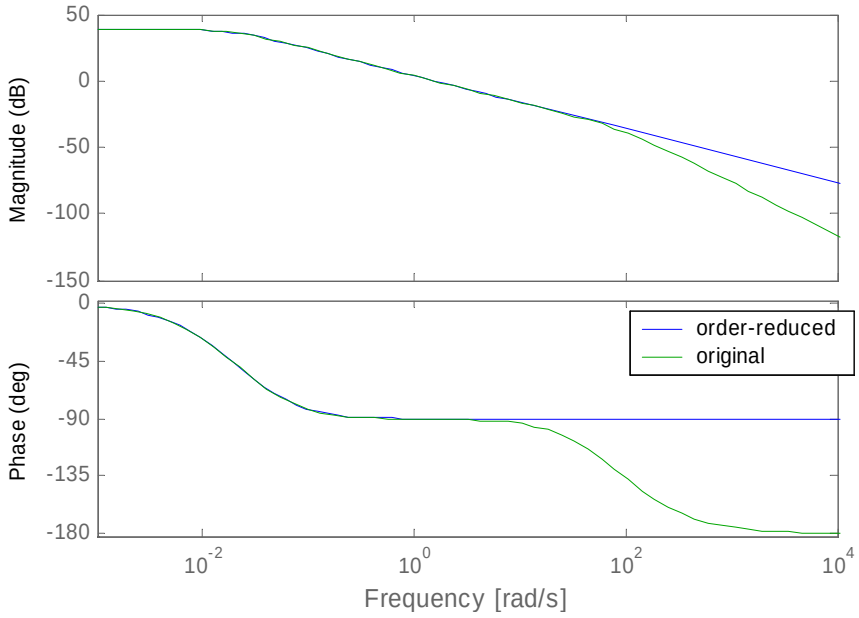

Figure 10: Bode plot of the transfer function $G_1(s)$ of (2.64) and the Bode plot of the order-reduced transfer function $G_{red,1}(s)$ of (2.69).

Now the controller is designed on the basis of the order-reduced transfer function. Since this transfer function is SPR, there is no need to extend the matching error as shown in (2.43) which simplifies the algorithm of the adaptation considerably. For a first simulation we choose the (tuning) parameters as

$$\begin{aligned} &G_m(s) = \frac{1}{s+1},\ \Lambda(s) = (s+1) \\ &\Gamma = diag\{0.05, 0.05, 0.05, 0.05, 0.05\} \end{aligned} \tag{2.70}$$

thus the matching parameters for the no-failure case are calculated as

$$\begin{aligned} &\theta_1^* = 0.847044,\ \theta_2^* = 0.186596,\ \theta_0^* = -0.405496, \\ &k^* = 0.21946,\ \theta_4^* = 0 \end{aligned} \tag{2.71}$$

For the simulations shown in Figure 11 an initial error of the matching parameter of +50% each is assumed, which results in a tracking error during the step. The actuator failures result in a transient response of the pitch attitude. Figure 12 depicts the adaptation of the controller parameters.

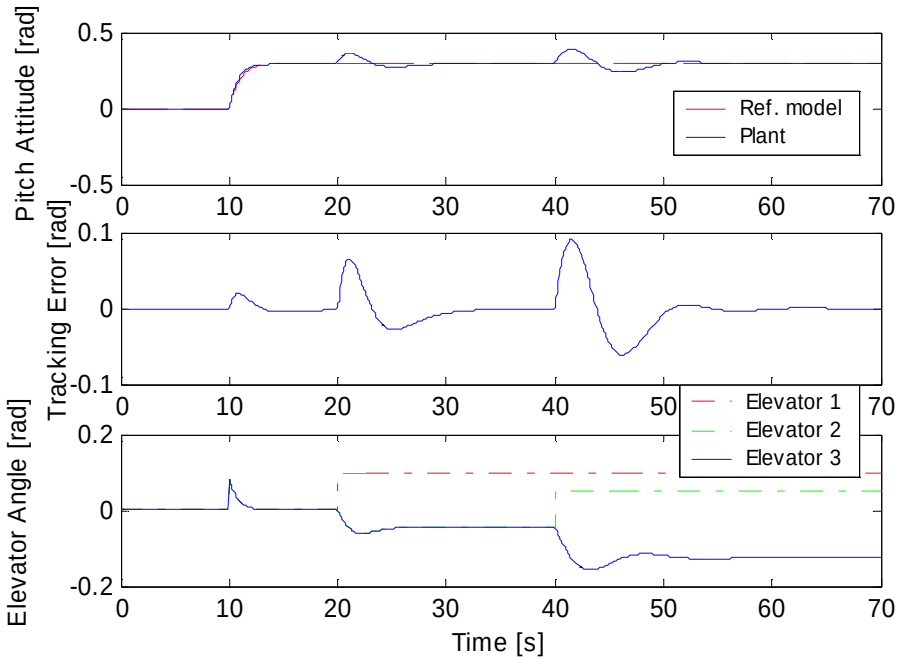

Figure 11: Simulated step of pitch attitude (with nonlinear model). Elevator 1 fails after 20s and Elevator 2 after 40s.

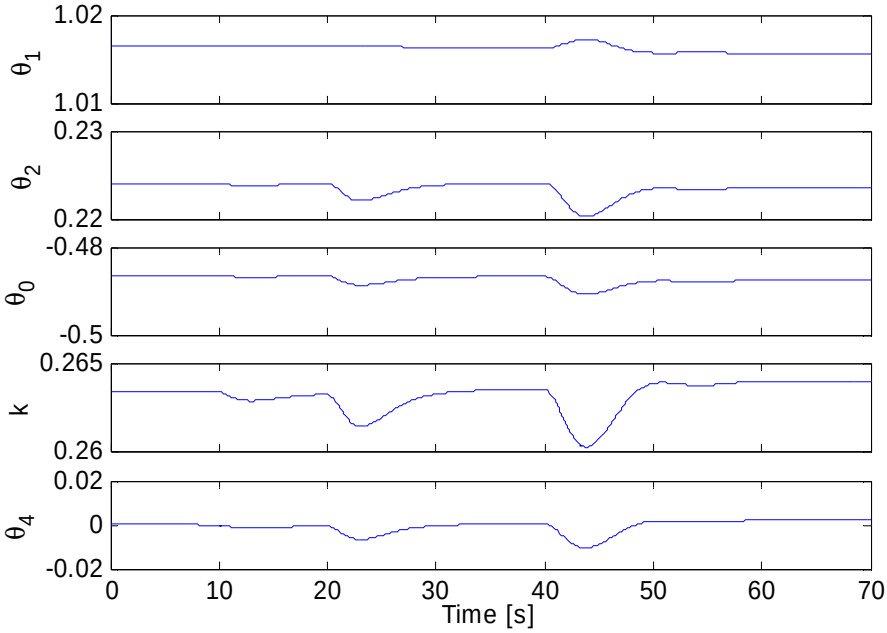

Figure 12: Controller parameters

For the next simulation the adaptation gain matrix is increased

$$\Gamma = diag\{0.15, 0.15, 0.15, 0.15, 0.15\} \tag{2.72}$$

Figure 13 shows that this results in a reduced damping of the transient response of the pitch attitude after an actuator failure. The same effect can be observed in the adaptation of the controller parameters in Figure 14. In order to compensate for the lack of ample damping the bandwidths of the auxiliary signal generator are increased. For the simulation in Figure 15 the denominator of the new auxiliary signals generator transfer function is therefore chosen as

$$\Lambda(s) = (s + 10) \tag{2.73}$$

Again assuming the no-failure case, the new matching parameters thus are

$$\begin{aligned} &\theta_1^* = 9.84704,\ \theta_2^* = 21.6123,\ \theta_0^* = \text{-}2.38063, \\ &k^* = 0.21946,\ \theta_4^* = 0 \end{aligned} \tag{2.74}$$

Furthermore the adaptation gain matrix is increased again

$$\Gamma = diag\{0.2, 0.2, 0.2, 0.2, 0.2\} \tag{2.75}$$

These new (tuning) parameters lead to a considerably improved compensation of the actuator failures. In the last simulation shown in this chapter, the pitch attitude is to follow the reference sine signal $r(t) = 0.0873\sin(0.1t)$. For the parameters we again use (2.73) and (2.75). The results of the simulation are depicted in Figure 17 and 18.

In this section it was shown that an appropriate choice of the process model is crucial for the performance of the MRAC method. The relevant dynamics should be characterized by the model; however, in order to reduce the number of parameters to be adapted the order of the model is to be kept as small as possible.

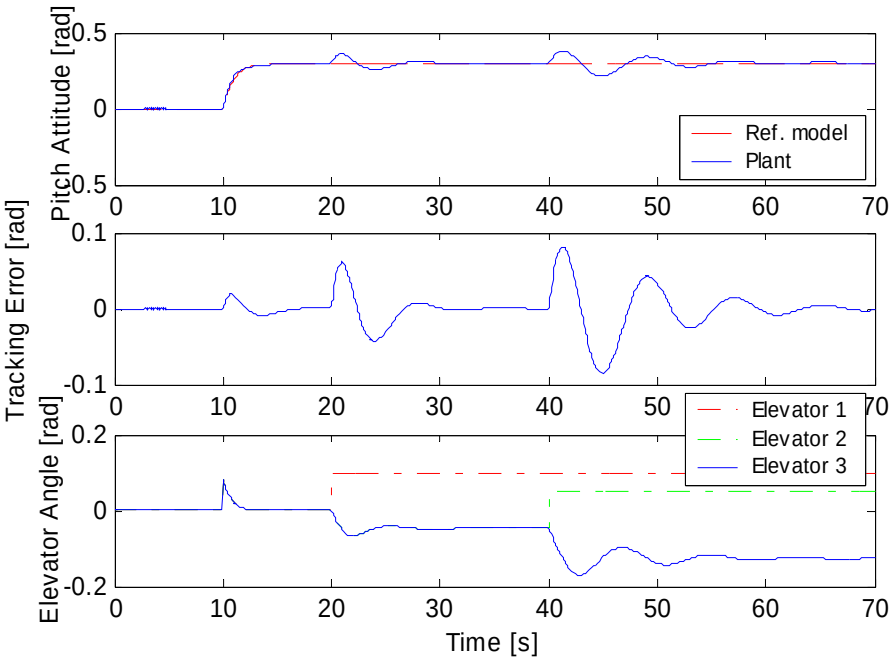

Figure 13: Simulated step of pitch attitude (with nonlinear model). Elevator 1 fails after 20s and Elevator 2 after 40s. New adaptation gain: $\Gamma = diag\{0.15, 0.15, 0.15, 0.15, 0.15\}$

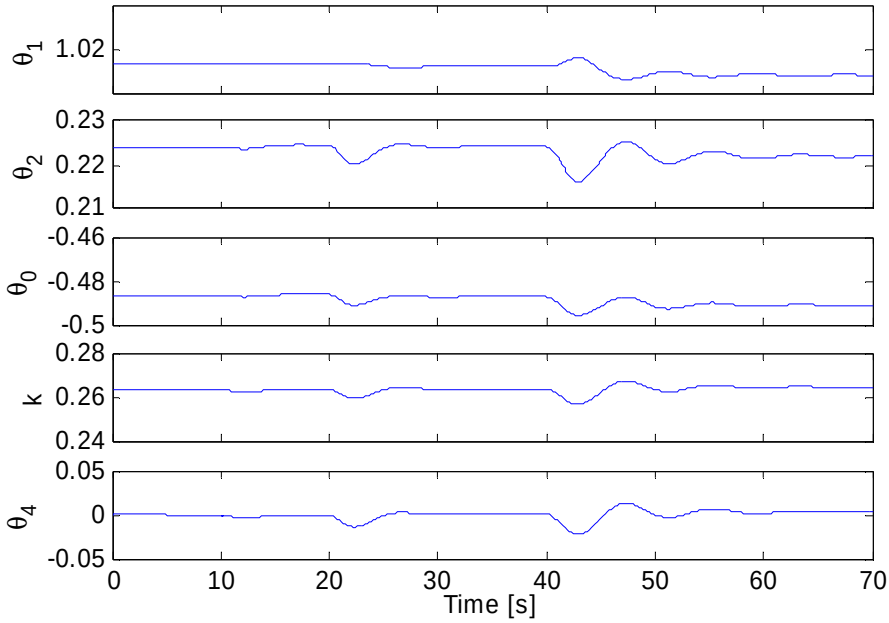

Figure 14: Controller parameters

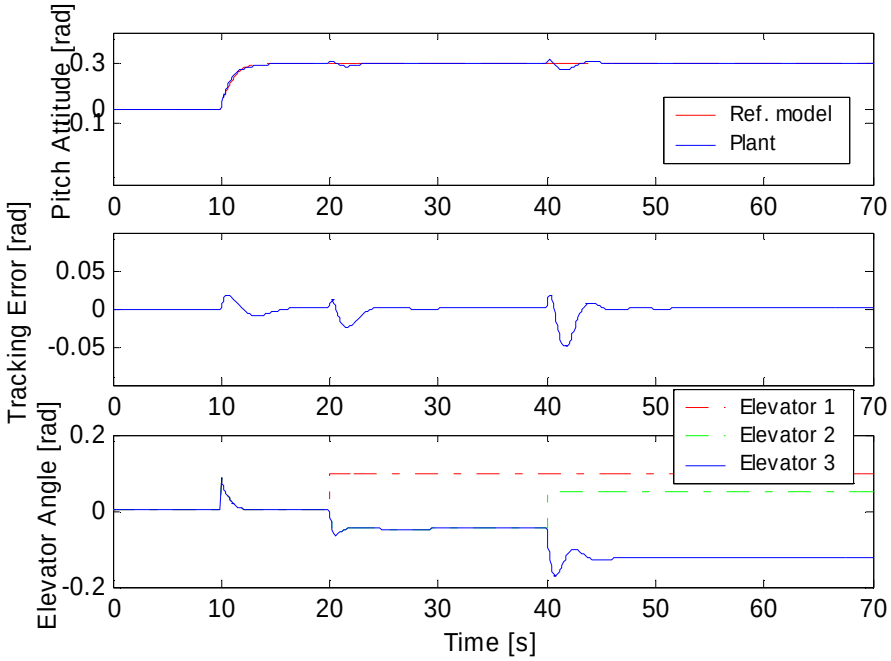

Figure 15: Simulated step of pitch attitude (with nonlinear model). Elevator 1 fails after 20s and Elevator 2 after 40s. The bandwidth of the auxiliary signal generator is increased according to (2.73). The adaptation gain is chosen as: $\Gamma = diag\{0.2, 0.2, 0.2, 0.2, 0.2\}$

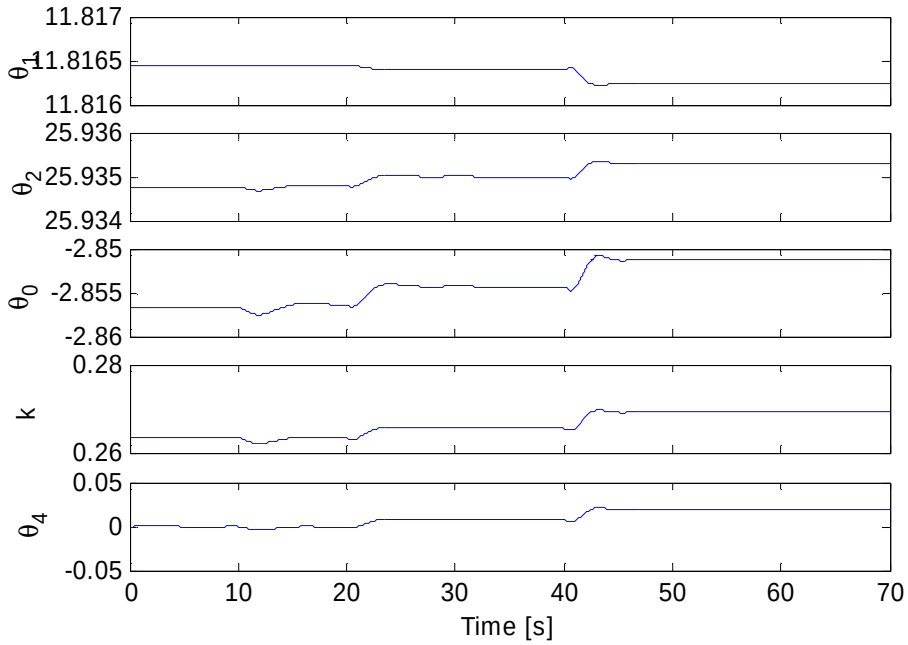

Figure 16: Controller parameters

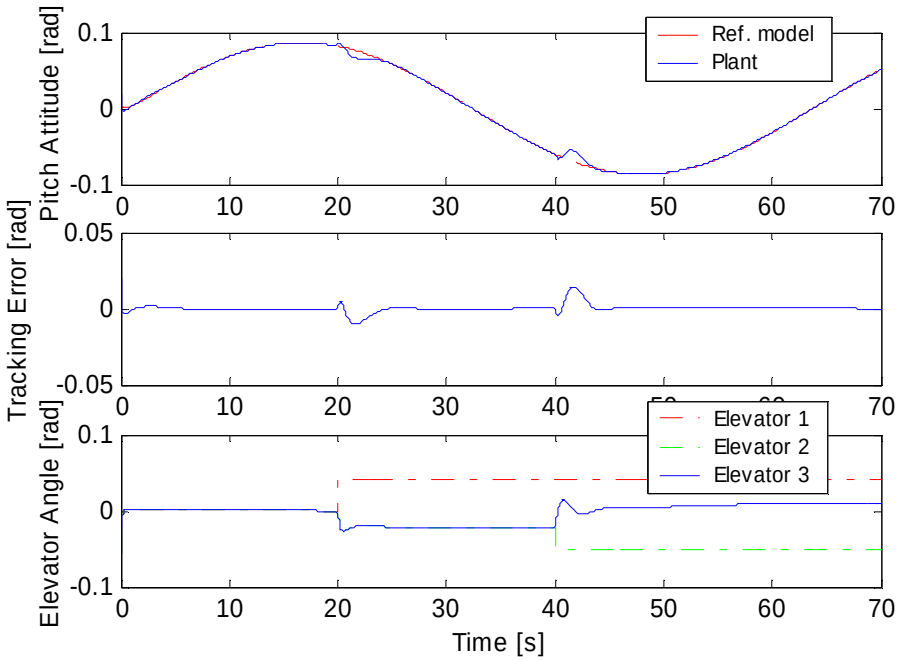

Figure 17: Simulated sine function of pitch attitude (with nonlinear model). Elevator 1 fails after 20s and Elevator 2 after 40s. The bandwidth of the auxiliary signal generator is increased according to (2.73). The adaptation gain is chosen as: $\Gamma = diag\{0.2, 0.2, 0.2, 0.2, 0.2\}$

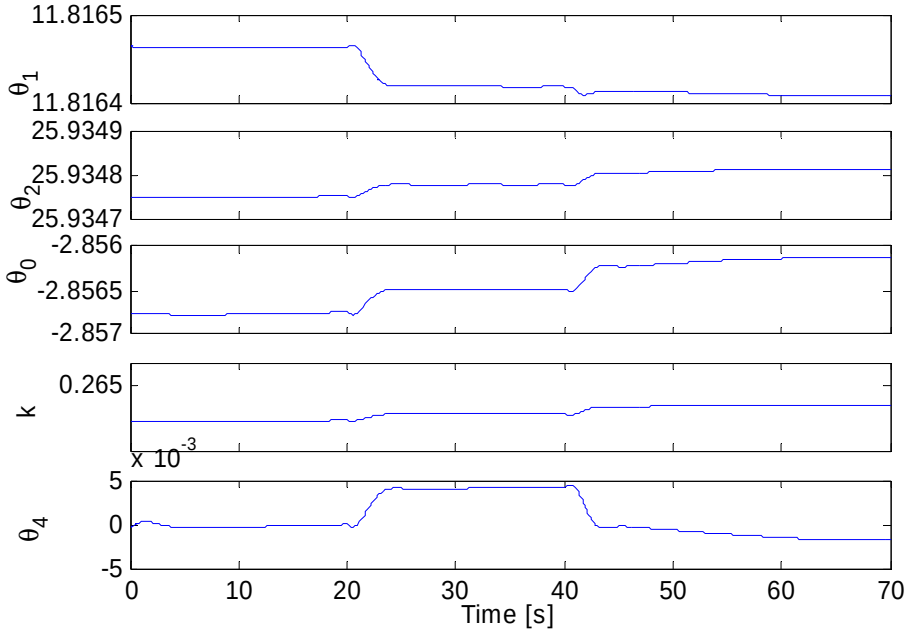

Figure 18: Controller parameters

3 Multiple Model Adaptive Estimation

In this chapter the multiple model adaptive estimation (MMAE) approach is used to detect sensor and actuator failures. The MMAE method is based on a parallel bank of Kalman filters, each of which is matched to a specific hypothesis about the failure status of the system (fully functional or a failure in any one sensor or actuator). The conditional probability of each hypothesis based on the measurement history is calculated in order to compute the probability-weighted state estimation of each Kalman filter, which can then be added up to an estimation of the state vector of the (failed) plant (see (3.1)). Hence the MMAE method provides information about the probability of a possible failure on the one hand, and on the other hand, an estimation of the state vector which may then be used for state feedback control.

$$\hat{x}(t) = p_{nf}\hat{x}_{nf} + p_{f1}\hat{x}_{f1} + \ldots + p_{fK}\hat{x}_{fK} \tag{3.1}$$

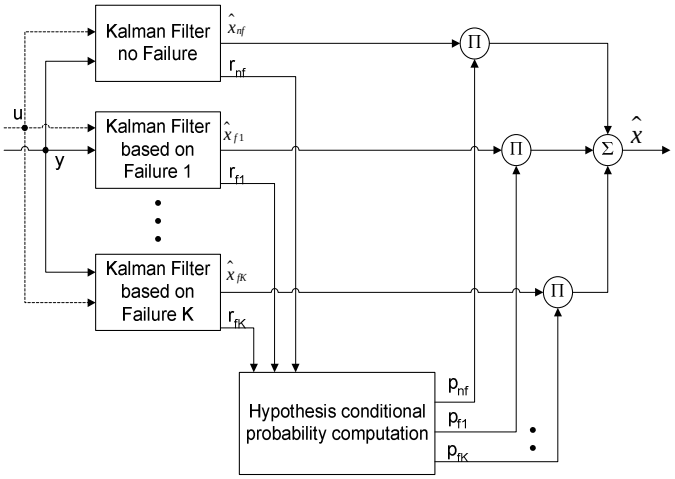

Figure 19: A Scheme of the MMAE algorithm based on a bank of Kalman filters and a hypothesis conditional probability computation.

3.1 The Kalman Filter Algorithm

Consider the system with the known parameter vector θ

$$\begin{aligned} x(t+1) &= F(\theta)x(t) + G_u(\theta)u(t) + G_v(\theta)v(t) \\ x(0) &= \xi \\ y(t) &= H(\theta)x(t) + r(t) \end{aligned} \tag{3.2}$$

and with its known expectations of initial state, model input, and measurement errors

$$\begin{aligned} &E\{\xi\} = \bar{\xi} \\ &E\{v(t)\} = 0 \qquad \text{for t} \geq 0 \\ &\mathrm{E}\{\mathrm{r(t)}\} = 0 \qquad \text{for t} \geq 0 \end{aligned} \tag{3.3}$$

and the initial covariance matrix of the estimation error as well as the auto-covariance matrix of both system and measurement noise

$$\begin{aligned} &E\{[\xi - \bar{\xi}][\xi - \bar{\xi}]^T\} = \Sigma_0 \qquad \text{with } \Sigma_0 = \Sigma_0^T \geq 0 \\ &E\{[v(t)][v(\tau)]^T\} = R_v(\theta)\delta_{t\tau} \quad \text{with } R_v(\theta) = R_v^T(\theta) \geq 0 \quad \text{for t}, \tau \geq 0 \\ &E\{[r(t)][r(\tau)]^T\} = R_r(\theta)\delta_{t\tau} \quad \text{with } R_r(\theta) = R_r^T(\theta) > 0 \quad \text{for t}, \tau \geq 0 \\ &E\{[v(t)][r(\tau)]^T\} = R_{vr}(\theta)\delta_{t\tau} \\ &E\{[\xi - \bar{\xi}][r(t)]^T\} = 0 \\ &E\{[\xi - \bar{\xi}][v(t)]^T\} = 0 \end{aligned} \tag{3.4}$$

For our desired estimation we set the following two conditions:

$$E\{e(\mathrm{t+1}|\mathrm{t})\} = 0 \tag{3.5}$$

with $e(\mathrm{t+1}|\mathrm{t})$ being the state estimation error at time t+1, given the data at time t

$$\mathrm{e(t+1|t)} = x(\mathrm{t+1}) - \hat{x}(\mathrm{t+1|t}) \tag{3.6}$$

and for any $\Sigma_{subopt}(t+1|t)$

$$\Sigma_{subopt}(t+1|t) - E\{e(\mathrm{t+1|t})\mathrm{e}^{\mathrm{T}}(\mathrm{t+1|t})\} \geq 0 \tag{3.7}$$

The following Kalman algorithm complies with the given requirements

$$\begin{aligned} \hat{x}(t+1|t) &= F(\theta)\hat{x}(t|t-1) + G_u(\theta)u(t) + K(\theta)[y(t) - H(\theta)\hat{x}(t|t-1)] \\ \hat{x}(0|-1) &= \bar{\xi} \end{aligned} \tag{3.8}$$

with the covariance matrix:

$$\begin{aligned} \Sigma(t+1|t) &= F(\theta)\Sigma(t|t-1)F^T(\theta) + G_v(\theta)R_v(\theta)G_v^T(\theta) - K(\theta)Q(\theta)K^T(\theta) \\ \Sigma(0|-1) &= \Sigma_0 \end{aligned} \tag{3.9}$$

with the Kalman gain

$$K(t,\theta) = [F(\theta)\Sigma(t|t-1)H^T(\theta) + G_v(\theta)R_{vr}(\theta)]Q^{-1}(\theta) \tag{3.10}$$

and the residual covariance matrix

$$Q(\theta) = H(\theta)\Sigma(t|t-1)H^T(\theta) + R_r(\theta) \tag{3.11}$$

For implementation purposes the equations above can be decomposed into two more convenient steps.

Data Update:

$$\begin{aligned}\hat{x}(t|t) &= \hat{x}(t|t-1) + L(t)[y(t) - H(\theta)\hat{x}(t|t-1)] \\ \hat{x}(0|-1) &= \bar{\xi}\end{aligned} \tag{3.12}$$

$$\begin{aligned}\Sigma(t|t) &= [I - L(t)H(\theta)]\Sigma(t|t-1)[I - L(t)H(\theta)]^T + L(t)R_r(\theta)L^T(t) \\ \Sigma(0|-1) &= \Sigma_0\end{aligned} \tag{3.13}$$

$$L(t) = \Sigma(t|t-1)H^T(\theta)Q^{-1}(t) \tag{3.14}$$

$$Q(t) = H(\theta)\Sigma(t|t-1)H^T(\theta) + R_r(\theta) \tag{3.15}$$

Prediction:

$$\hat{x}(t+1|t) = F(\theta)\hat{x}(t|t) + G_u(\theta)u(t) + G_v(\theta)R_{vr}(\theta)Q^{-1}(\theta)[y(t) - H(\theta)\hat{x}(t|t)] \tag{3.16}$$

$$\begin{aligned}\Sigma(t+1|t) = {} & F(\theta)\Sigma(t|t)F^T(\theta) + G_v(\theta)R_v(\theta)G_v^T(\theta) \\ & -G_v(\theta)R_{vr}(\theta)Q^{-1}(t)R_{vr}^T(\theta)G_v^T(\theta) \\ & -F(\theta)L(t)R_{vr}^T(\theta)G_v^T(\theta) \\ & -G_v(\theta)R_{vr}(\theta)L^T(\theta)F^T(\theta)\end{aligned} \tag{3.17}$$

3.2 Hypothesis Testing

Given a dynamic system with K possible failure patterns which may occur during operation, let θ denote the vector of (uncertain) parameters depicting the failure status of sensors and actuators of the system. Assume that all K possible conditions can be described by discrete values θ_k of θ (for k = 1, 2, ..., K). Assume further that a bank of K separate Kalman filters is given (see Figure 19), each filter representing a hypothesis for a possible failure pattern of the system. Now we define the hypothesis-conditional probability $p_k(t_i)$ as the probability that θ assumes the value θ_k conditioned on the observed measurement history $Y(t_i)$ at time t_i.

$$p_k(t_i) = P\left[\theta = \theta_k \middle| Y(t_i) = Y_i\right] \tag{3.18}$$

In the next step a recursive expression for the conditional probability in (3.18) is derived. For the following calculations, some results of conditional probability theory [JAZ-70] are presented

The conditional density function $f_{X|Y}(x|y)$ of X given $\{Y(\omega) = y\}$ for all x and y such that the marginal density $f_Y(y) = \int f_{X,Y}(x, y)dx > 0$ is defined by

$$f_{X|Y}(x|y) \triangleq \frac{f_{X,Y}(x, y)}{f_Y(y)} \tag{3.19}$$

Note that the roles of X and Y can be reversed which gives the equivalent result

$$f_{Y|X}(y|x) = \frac{f_{X,Y}(x, y)}{f_X(x)} \tag{3.20}$$

Furthermore the well-known Bayes' rule can be written in "density form"

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)\, f_X(x)}{f_Y(y)} \tag{3.21}$$

With these prerequisites the conditional density function $f_{\theta|Y(t_i)}(\theta_k \,|Y_i)$ of θ given $\{Y(t_i) = Y_i\}$ for all θ_k and Y_i can be written as

$$f_{\theta|Y(t_i)}(\theta_k \,|Y_i) = \frac{f_{\theta,Y(t_i)}(\theta_k, Y_i)}{f_{Y(t_i)}(Y_i)} \tag{3.22}$$

where the marginal density is given by

$$f_{Y(t_i)}(Y_i) = \int f_{\theta,Y}(\vartheta, Y_i)\, d\vartheta = \sum_{j=1}^{K} f_{\theta,Y(t_i)}(\theta_j, Y_i) > 0 \tag{3.23}$$

Equation (3.22) can be rewritten as

$$f_{\theta|Y(t_i)}(\theta_k \,|Y_i) = \frac{f_{\theta,Y(t_{i-1}),y(t_i)}(\theta_k, Y_{i-1}, y_i)}{f_{Y(t_i)}(Y_i)} \tag{3.24}$$

Now with (3.19) the joint density function $f_{\theta,Y(t_{i-1}),y(t_i)}(\theta_k, Y_{i-1}, y_i)$ can be written as the product of the probability density function of the current measurement y_i, conditioned on the particular assumed parameter value θ_k and the observed past measurement history Y_{i-1}: $f_{y(t_i)|\theta,Y(t_{i-1})}(y_i \,|\theta_k, Y_{i-1})$ and the joint density function $f_{\theta,Y(t_{i-1})}(\theta_k, Y_{i-1})$.

$$\begin{aligned} f_{\theta,Y(t_i)}(\theta_k, Y_i) = f_{Y(t_i),\theta}(Y_i, \theta_k) &= f_{y(t_i),Y(t_{i-1}),\theta}(y_i, Y_{i-1}, \theta_k) \\ &= f_{y(t_i)|\theta,Y(t_{i-1})}(y_i \,|\theta_k, Y_{i-1})\, f_{\theta,Y(t_{i-1})}(\theta_k, Y_{i-1}) \end{aligned} \tag{3.25}$$

Hence (3.24) can be written as

$$f_{\theta|Y(t_i)}(\theta_k \,|Y_i) = \frac{f_{y(t_i)|\theta,Y(t_{i-1})}(y_i \,|\theta_k, Y_{i-1})\, f_{\theta,Y(t_{i-1})}(\theta_k, Y_{i-1})}{f_{Y(t_i)}(Y_i)} \tag{3.26}$$

Now with equation (3.25), the right side of equation (3.23) is given by

$$\sum_{j=1}^{K} f_{\theta,Y(t_i)}(\theta_j, Y_i) = \sum_{j=1}^{K} f_{y(t_i)|\theta,Y(t_{i-1})}(y_i \,|\theta_j, Y_{i-1})\, f_{\theta,Y(t_{i-1})}(\theta_j, Y_{i-1}) \tag{3.27}$$

Therefore equation (3.26) may be written as

$$f_{\theta|Y(t_i)}(\theta_k \,|Y_i) = \frac{f_{y(t_i)|\theta,Y(t_{i-1})}(y_i \,|\theta_k, Y_{i-1})\, f_{\theta,Y(t_{i-1})}(\theta_k, Y_{i-1})}{\sum_{j=1}^{K} f_{y(t_i)|\theta,Y(t_{i-1})}(y_i \,|\theta_j, Y_{i-1})\, f_{\theta,Y(t_{i-1})}(\theta_j, Y_{i-1})} \tag{3.28}$$

Since the discrete conditional density function can be directly related to the conditional probability with

$$f_{\theta|Y(t_i)}(\vartheta|Y_i) = \sum_{k=1}^{K} p_k(t_i)\delta(\vartheta - \theta_k) \tag{3.29}$$

where δ is the Dirac function (e.g. $f_{\theta|Y(t_i)}(\theta_j \,|Y_i) = \sum_{k=1}^{K} p_k(t_i)\delta(\theta_j - \theta_k) = p_j(t_i)$), equation (3.28) can be rewritten as

$$p_k(t_i) = \frac{f_{y(t_i)|\theta,Y(t_{i-1})}(y_i \,|\theta_k, Y_{i-1})\, p_k(t_{i-1})}{\sum_{j=1}^{K} f_{y(t_i)|\theta,Y(t_{i-1})}(y_i \,|\theta_j, Y_{i-1})\, p_j(t_{i-1})} \tag{3.30}$$

which is a recursive representation of (3.26) and thus can be used in an “online” algorithm. The conditional density $f_{y(t_i)|\theta,Y(t_{i-1})}(y_i \,|\theta_k, Y_{i-1})$ can be computed with the information provided by its assigned Kalman filter

$$f_{y(t_i)|\theta,Y(t_{i-1})}\left(y_i \,|\theta_k, Y_{i-1}\right) = \frac{1}{(2\pi)^{m/2}\det\left(Q_k(t_i)\right)^{1/2}} \exp\left(-\frac{1}{2} r_k^T(t_i) Q_k^{-1}(t_i) r_k(t_i)\right) \tag{3.31}$$

where m is the measurement dimension, $r_k(t_i) = y_k(t) - H_k(\theta)\hat{x}_{fk}(t|t-1)$ is the residual, and $Q_k(t_i)$ is the residual covariance matrix of the k$^{\text{th}}$ filter at time t_i.

3.3 Implementation in Matlab/Simulink

The MMAE algorithm has been implemented in Matlab/Simulink. For the large bundle of signals which emanate from the Kalman filter bank, a bus system is used for a more convenient handling. The bus creator given in Simulink was used to summon the different signals (e.g. vector and matrix signals) into one bus system. Now all the Kalman magnitudes can be "transported" in one single signal thread. Finally the bus selector can be used to extract the desired signal.

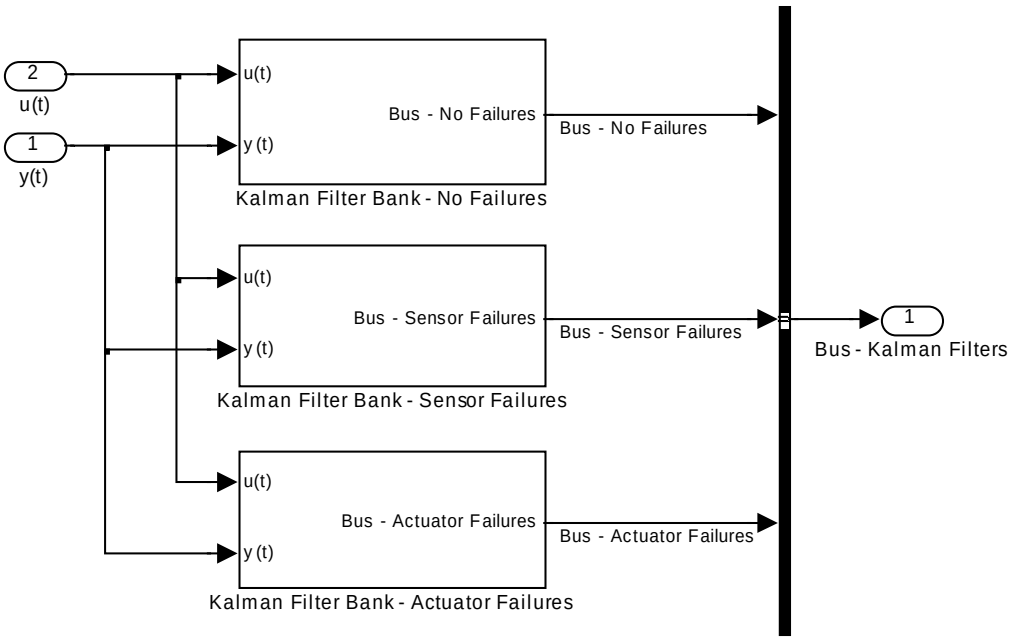

Figure 20: Implementation of the Kalman filter bank in Matlab/Simulink.

3.4 Modeling of Actuator and Sensor Failures

This section describes the general nature of failures which can be detected with the MMAE algorithm. Based on each possible failure hypothesis of the system, an appropriate Kalman filter needs to be designed, i.e., the prior information of each failure is used to compute compatible linear models which are then used for the design of the respective Kalman filter. It has to be emphasized that the MMAE method can only detect failures which are taken into account by a predefined Kalman filter (hypothesis). Hence all information about the manner of a specific failure needs to be known. This is a very restrictive requirement since most failures are not fully known in advance (e.g. a rudder of an airplane may be stuck at an unknown position, which would imply that for every rudder position possible a separate Kalman filter needs to be designed). In this section we deal with "hard" failures, i.e., for actuators we assume that in case of a failure the actuator will have no further effects on the system (e.g., a rudder being stuck at zero angle), furthermore a failed sensor only produces unbiased (white) measurement noise. With these assumptions it is easy to derive a (linear) model of the failed plant, thus a single actuator failure can be represented by setting its associated column in the input matrix to zero and a single sensor failure by setting its associated row in the measurement matrix to zero.

Consider the (healthy) system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{3.32}$$

with $A \in \mathbb{R}^{n\times n}, B = \left[b_1,...,b_m\right] \in \mathbb{R}^{n\times m}$ and $C = \left[c_1,...,c_p\right]^T \in \mathbb{R}^{p\times n}$. In case of a hard failure of actuator q, $1 \le q \le m$ the input matrix needs to be rewritten as $B = \left[b_1,...,b_q = 0,...,b_m\right]$ and in case of a full failure of sensor r, $1 \le r \le p$ the measurement matrix needs to be rewritten as $C = \left[c_1,...,c_r = 0,...,c_p\right]^T$.

3.5 Simulation Results

The failure detection with the MMAE method was tested on the nonlinear model aircraft model of tenth order described in chapter 5. For test purposes two groups of redundant actuators (two ailerons and two rudders) and two groups of redundant sensors (two roll rate sensors and two yaw rate sensors) are added. For the following simulations, we assume that only the redundant components may fail. For the design of the Kalman filters the nonlinear model needs to be linearized. The linearization was computed analytically with Mathematica. The set point is chosen to be an equilibrium point

$$x_{nom}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \\ x_7(t) \\ x_8(t) \\ x_9(t) \\ x_{10}(t) \end{bmatrix}_{nom} \triangleq \begin{bmatrix} p(t) \\ q(t) \\ r(t) \\ q_0(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ u(t) \\ v(t) \\ w(t) \end{bmatrix}_{nom} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 38.3988 \\ 0 \\ 1.08312 \end{bmatrix} \begin{matrix} \text{roll rate } [\text{rad/s}] \\ \text{nick rate } [\text{rad/s}] \\ \text{yaw rate } [\text{rad/s}] \\ \text{Euler parameter } [-] \\ \text{"} \\ \text{"} \\ \text{"} \\ \text{longitudinal velocity } [\text{m/s}] \\ \text{lateral velocity } [\text{m/s}] \\ \text{normal velocity } [\text{m/s}] \end{matrix} \tag{3.33}$$

The inputs are

$$u_{nom}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \\ u_5(t) \\ u_6(t) \end{bmatrix}_{nom} \triangleq \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \eta(t) \\ \zeta_1(t) \\ \zeta_2(t) \\ f(t) \end{bmatrix}_{nom} = \begin{bmatrix} 0 \\ 0 \\ 0.0174073 \\ 0 \\ 0 \\ 50 \end{bmatrix} \begin{matrix} \text{aileron 1 angle [rad]} \\ \text{aileron 2 angle [rad]} \\ \text{elevator angle [rad]} \\ \text{rudder 1angle [rad]} \\ \text{rudder 2 angle [rad]} \\ \text{thrust force [N]} \end{matrix} \tag{3.34}$$

Note that the redundant actuators are assumed to be “cloned” actuators, i.e., the ways in which redundant input signals “enter” the system only differ by an “efficiency” factor (see Section 5.2).

The outputs are chosen as

$$y(t)=\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \\ y_5(t) \\ y_6(t) \\ y_7(t) \\ y_8(t) \end{bmatrix} \triangleq \begin{bmatrix} p(t) \\ p(t) \\ q(t) \\ r(t) \\ r(t) \\ u(t) \\ v(t) \\ w(t) \end{bmatrix} \begin{matrix} \text{roll rate sensor 1 } [\text{rad/s}] \\ \text{roll rate sensor 2} [\text{rad/s}] \\ \text{nick rate } [\text{rad/s}] \\ \text{yaw rate sensor 1 } [\text{rad/s}] \\ \text{yaw rate sensor 2 } [\text{rad/s}] \\ \text{longitudinal velocity } [\text{m/s}] \\ \text{lateral velocity } [\text{m/s}] \\ \text{normal velocity } [\text{m/s}] \end{matrix} \tag{3.35}$$

The resulting system matrices are listed in Appendix A.3. Simulations show that the hypothesis testing algorithm works more reliably when the linear system is normalized properly so that all orders of magnitude are equal. The normalization matrices used for the following simulation can also be found in Appendix A.3. For the Kalman filters the time continuous matrices are discretized with the Euler forward rule

$$\dot{x}(t) \approx \frac{x(t+t_s)-x(t)}{t_s} \tag{3.36}$$

where the sample time for the following simulations was chosen as $t_s = 0.005s$. The discrete system matrices are

$$\begin{aligned} F &= t_s A + I_n \\ G &= t_s B \\ H &= C \end{aligned} \tag{3.37}$$

For the Kalman filters we define the following parameters, the initial covariance matrix of the estimation error and the auto-covariance matrix of both system and measurement noise

$$\begin{aligned} \Sigma_0 &= 1 I_n, \\ R_v &= 0.01 I_m, \\ R_r &= 1 I_p \end{aligned} \tag{3.38}$$

The measurement noise of the rate sensors is chosen as a normally distributed random signal with a variance of 0.0004 rad/s, the input noise of the ailerons and rudders is assumed to have a variance of 0.0001 rad/s. For the hypothesis testing we need to define initial probability values. In order to keep the probability calculation lively the probabilities should not go below

a given limit, otherwise, due to numerical errors, certain probabilities may go (exactly) to zero and then from (3.30) it can be seen that those probabilities stay at zero forever. For the following simulations a lower limit of 0.001 is used. Since we assume that at the beginning there is no failure, we choose for the initial values

$$\begin{aligned} p_{Af1} &= 0.001,\ p_{Af2} = 0.001 \\ p_{Af3} &= 0.001,\ p_{Af4} = 0.001 \\ p_{Sf1} &= 0.001,\ p_{Sf2} = 0.001 \\ p_{Sf3} &= 0.001,\ p_{Sf4} = 0.001 \end{aligned} \tag{3.39}$$

where p_{Afi} is the probability that the ith actuator fails and p_{Sfj} is the probability that the jth sensor fails, the probability that no failure occurs can then be written as

$$p_{nofailure} = 1 - \sum_{i=1}^{4} p_{fAi} - \sum_{i=j}^{4} p_{fSj} = 0.992 \tag{3.40}$$

Failure detection and isolation using the MMAE algorithm requires a stimulus to disturb the system from a quiescent state. The performance depends upon the magnitude of the residuals within incorrect filters having large residual values. Small deviations from a quiescent state will be virtually indistinguishable from system noise, providing poor detection and identification.

In the first simulation an aileron failure scenario is investigated. In order to show the influence of the exciting signal on the failure detection and isolation we choose a low-frequency sine signal as input to the ailerons $u_1(t) = u_2(t) = 0.0873 \sin(0.5t)$. At 4.5 s aileron 1 fails. Figure 22 shows an immediate detection of the failure; however, since both ailerons get exactly the same (low frequent) signal the computation of the probability encounters difficulties in distinguishing between the two (redundant) ailerons. In this case the only reason why a distinction is possible is the fact that the efficiency coefficients (5.16) of the two ailerons are not the same (c_1=1, c_2=0.6). The lower plot of Figure 21 shows the probability-weighted estimation of the roll rate; the substantial estimation error can also be attributed to the lack of sufficient excitation. Figure 23 and 24 show a failure scenario of the second aileron at 4.5 s with an input frequency of 1 rad/s (note that the first aileron has a larger efficiency coefficient and therefore the effect of the failure is not as significant as in the first simulation); the estimation of the roll rate as well as the probability estimation are considerably improved.

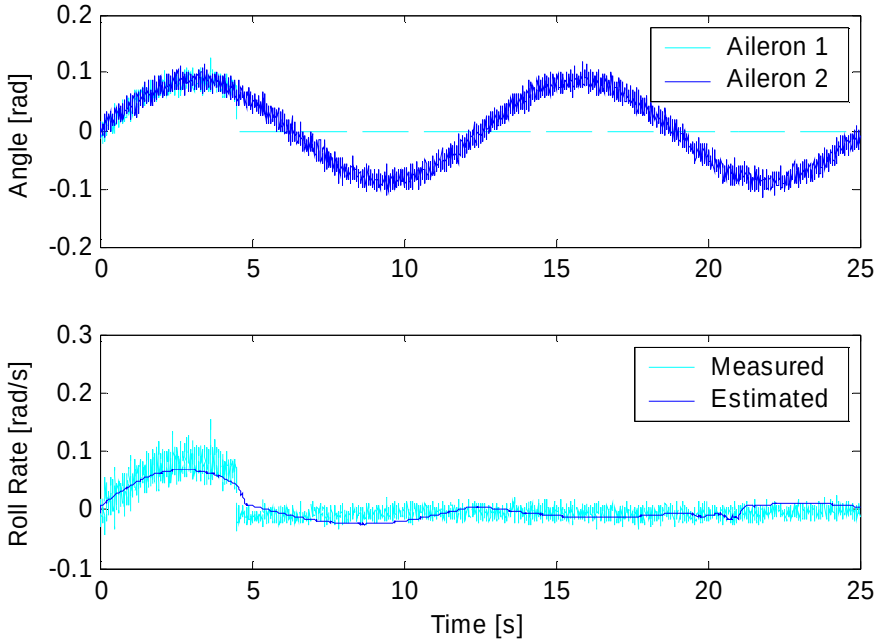

Figure 21: Aileron input signal and roll rate. Hard failure of aileron 1 at 4.5 s. The estimation is not satisfactory due to insufficient excitation of the roll dynamics. Inputs: $u_1(t)=u_2(t)=0.0873\ \sin(0.5t)$, $u_3(t)=u_5(t)=u_6(t)=0$, $u_4(t)= 0.0873\ \sin(0.5t+\pi)$. Efficiency coefficients of ailerons: $c_1=1$, $c_2=0.6$.

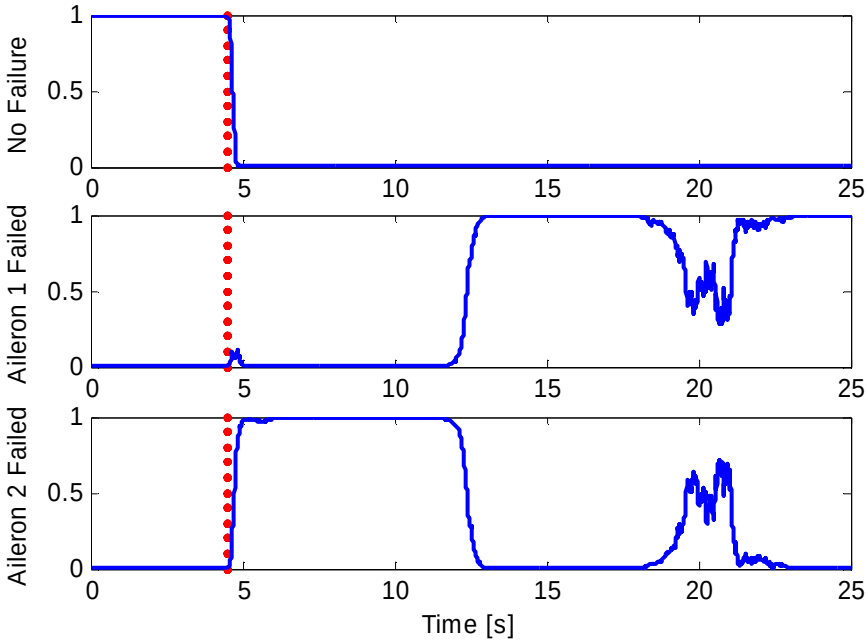

Figure 22: Probability that either no failure or a failure of aileron 1 or 2 has occurred. The dots indicate the time of the failure. Ambiguous distinction of redundant ailerons due to insufficient excitation.

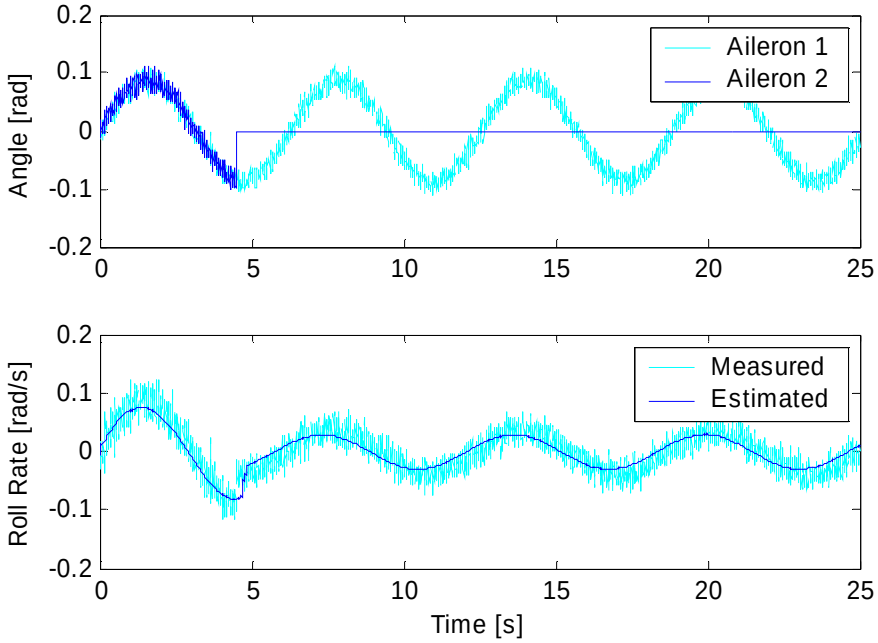

Figure 23: Aileron input signal and roll rate. Hard failure of aileron 2 at 4.5 s. The higher frequency of the input signal and the higher efficiency coefficient of aileron 1 leads to a better estimation of the roll rate. Inputs: $u_1(t)=.u_2(t)=0.0873$ sin(t), $u_3(t)=u_5(t)=u_6(t)=0$, $u_4(t)=0.0873$ sin(t+π). Efficiency coefficients of ailerons: $c_1=1$, $c_2=0.6$.

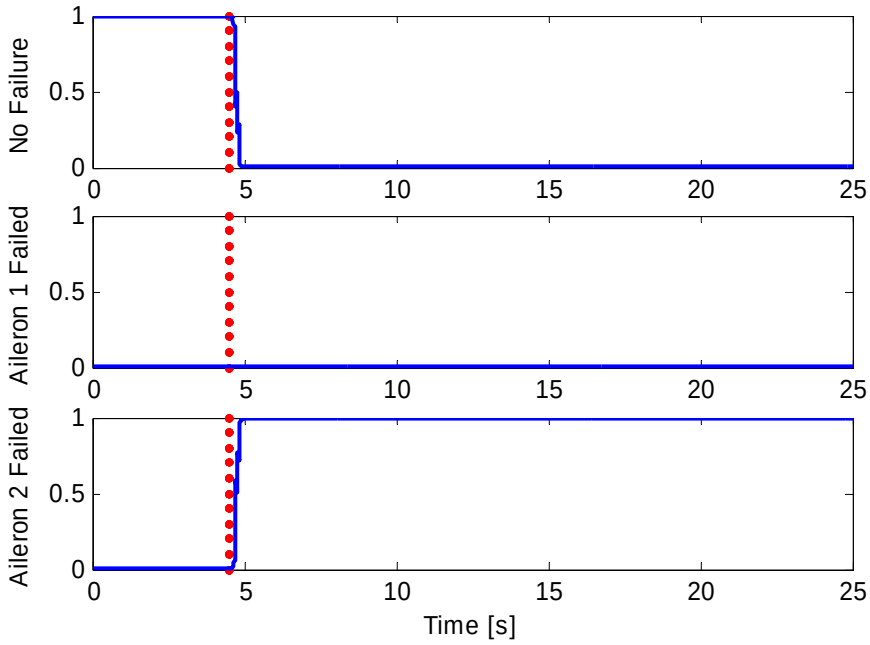

Figure 24: Probability that either no failure or a failure of aileron 1 or 2 has occurred. The failure is instantly detected and isolated.

On the sensor side detection and isolation of (hard) failures is easier since a failed sensor has a direct influence on the residuals of the Kalman filters.

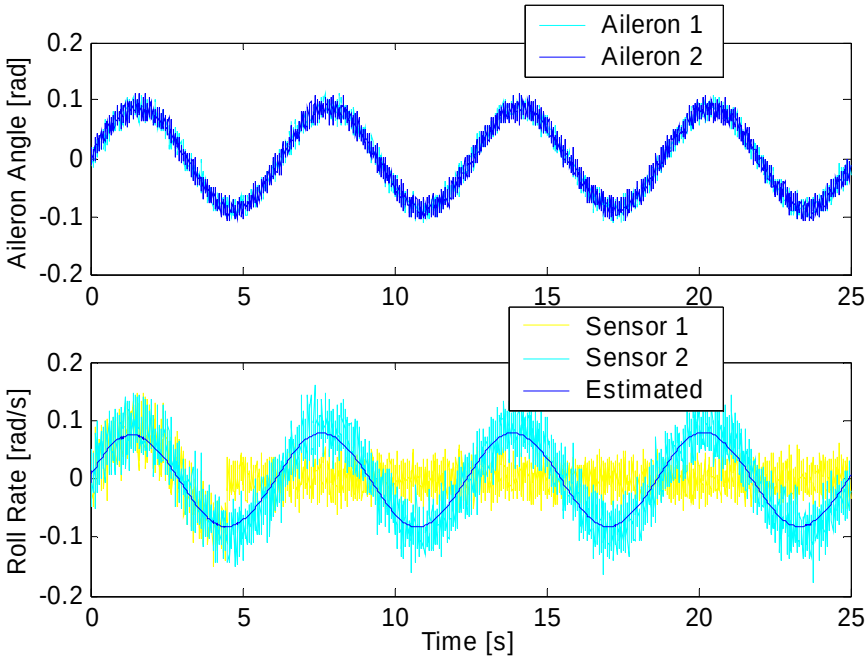

Figure 25: Aileron input signal and roll rate. Hard failure of roll rate sensor 1 at 4.5 s. Inputs: $u_1(t)=u_2(t)=0.0873\ \sin(t)$, $u_3(t)= u_6(t)=0$, $u_4(t)= u_5(t)=0.0873\ \sin(t+\pi)$. Efficiency coefficients of ailerons: $c_1=1$, $c_2=0.6$.

Figure 25 shows a scenario of a hard failure of the first roll rate sensor. After 4.5 s the subject sensor delivers zero mean white noise. According to Figure 26, the failure is immediately detected and isolated.

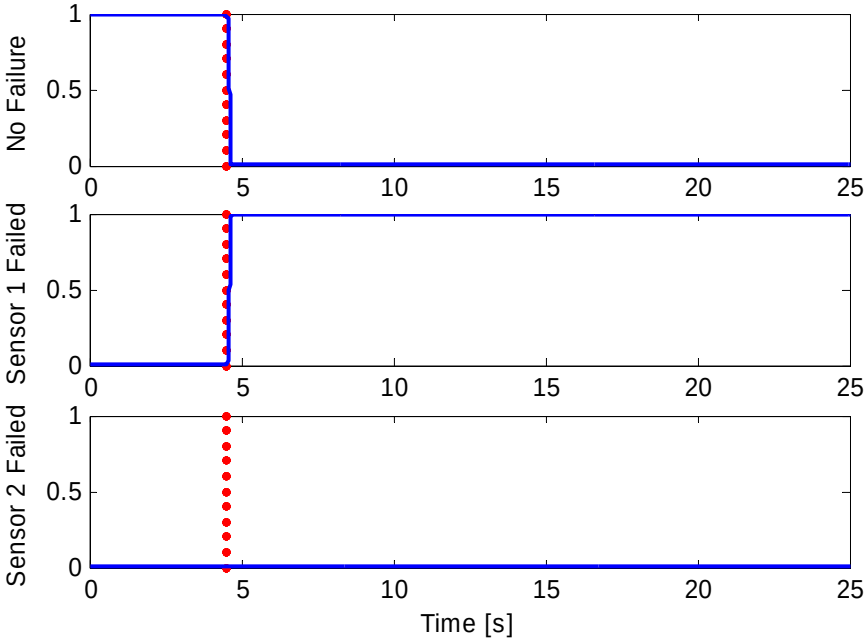

Figure 26: Probability that either no failure or a failure of roll rate sensor 1 or 2 has occurred. The dots indicate the time of the failure.

However, if we assume that either on the actuator or on the sensor side lock-in-place failures may occur, i.e., the residuals of the Kalman filters are biased, the MMAE method reaches its limit. We know from Kalman filter theory that we have to make allowance for all systematic errors, however, as lock-in-place failures cannot be predicted they may have detrimental effects on the filter performance. Figure 27 and 28 show the possible outcome of the MMAE failure detection in case of a lock-in-place failure. Due to the biased residual, the Kalman filter gives a wrong estimation of the state variables, which leads to severe problems with the probability calculation. Figure 28 shows that a failure of aileron 1 instead of aileron 2 is detected, and that even a failure of rudder 1 is computed for a short time, and that the calculations switch between “failure” and “no failure”. Therefore, neither the failure detection nor the failure isolation works properly.

In the next chapter an extended MMAE method is presented which can cope with lock-in-place failures.

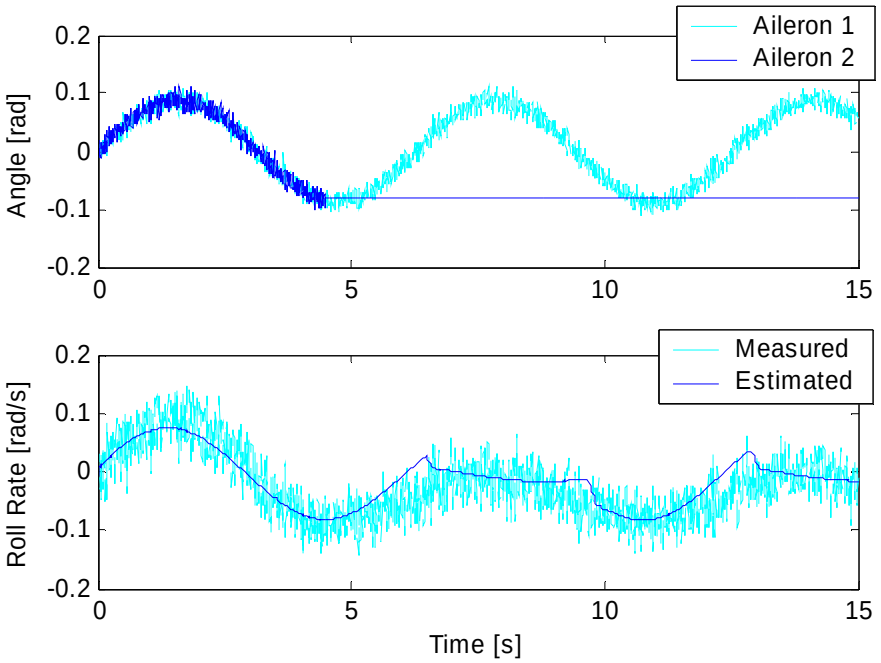

Figure 27: Aileron input signal and roll rate. Lock-in-place failure of aileron 2 at 4.5 s. The estimation is not satisfactory due to the biased input. Inputs: $u_1(t)=u_2(t)=0.0873\ \sin(t)$, $u_3(t)=u_6(t)=0$, $u_4(t)=u_5(t)=0.0873\sin(t+\pi)$. Efficiency coefficients of ailerons: $c_1=1$, $c_2=0.6$.

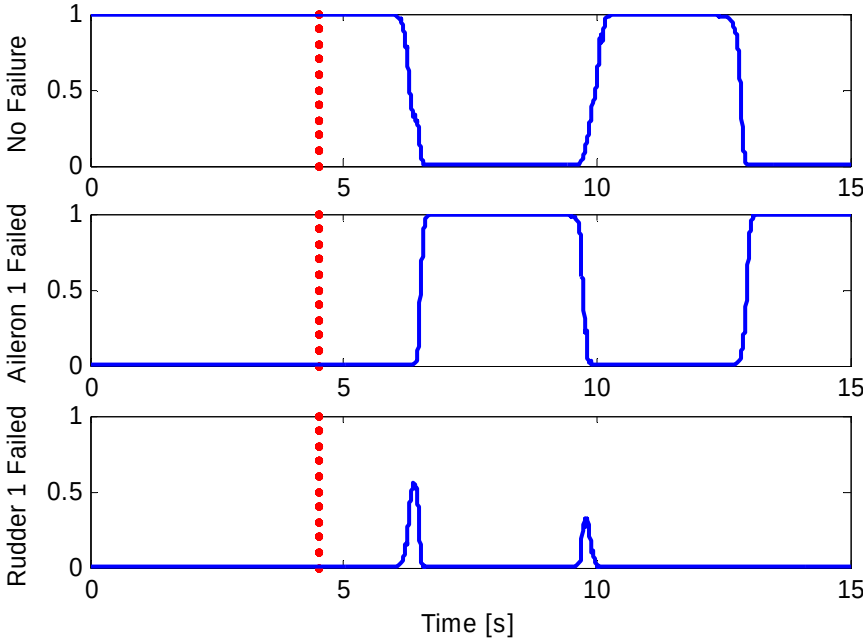

Figure 28: Probability that either no failure or a failure of aileron 1 or rudder 1 has occurred. The dots indicate the time of the failure. The lock-in-place failure leads to ambiguous probability calculations.

3.6 Multiple Failures

If we assume that two failures may happen during operation the MMAE method needs to be extended with new sets of Kalman filters incorporating dual failures. To reduce the number of filters required on line, a hierarchical approach, as shown in Figure 29, is employed. To begin

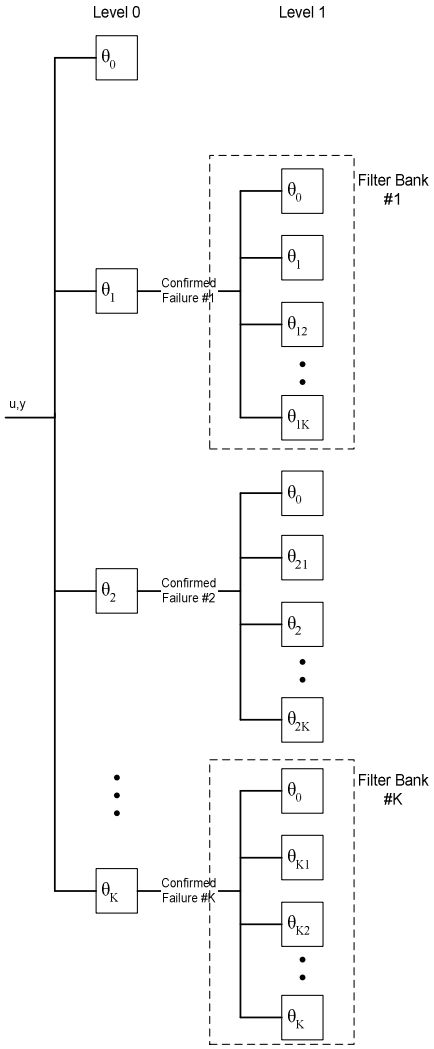

Figure 29: Hierarchical structure of Kalman filter bank. If a level 0 failure is confirmed, the respective level 1 bank becomes active.

with, only the K single-failure hypothesis filters are on line. Upon declaration of a failure, a new bank of filters is brought on line from memory storage. This bank contains filters designed for the declared failure, all dual failure combinations which include that failure (the doubly subscripted hypotheses in "Level 1" of Figure 29), and the no-failure hypothesis (to "back out" of the decision tree if necessary) [EID-96]. In this work no simulations were made with dual failures.

4 Extended Multiple Model Adaptive Estimation

In this chapter the MMAE algorithm is extended in order to make it accessible to lock-in-place failures and even to the more general class of varying failures (as long as the varying part is not too fast). The MMAE algorithm was found to be very powerful (only) for fully known (obviously except the point of time) failures; in the following the MMAE concept is combined with extended Kalman filters (EKF) which are able to estimate some (unknown) failure parameters. The resulting method is in this context called “extended multiple model adaptive estimation” (EMMAE).

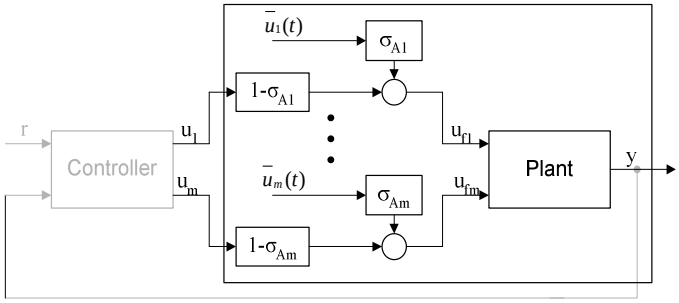

Figure 30: Control system with actuator failures

Consider again the control system with actuator failures described in Section 2.1 (depicted again in Figure 30). The actuator failures are modeled as

$$u_f(t) = u(t) + \sigma_A(\bar{u}(t) - u(t)) \tag{4.1}$$

where $u(t)$ is the desired plant input and

$$\bar{u}(t) = \left[\bar{u}_1(t), \bar{u}_2(t), ..., \bar{u}_m(t)\right]^T, \sigma_A = diag\left\{\sigma_{A1}, \sigma_{A2}, ..., \sigma_{Am}\right\} \tag{4.2}$$

with

$$\sigma_{Aj} = \begin{cases} 1 & \text{if the j}^{\text{th}}\text{ actuator fails} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

For the following investigation we assume that only one failure happens during the entire simulation period. We assume further that for every pair of $\bar{u}_i, \sigma_i$ a separate EKF was designed and used in a MMAE structure instead of the ordinary Kalman filters. The MMAE part of the EMMAE algorithm (see Figure 31) is now used to detect the switch of σ_j from zero to one and the EKF part is used to estimate the unknown actuator position $\bar{u}_j$ (assuming

that j is not known a priori). With this additional degree of freedom the EMMAE method can be used for all actuator failures which can be described by the combination of a hard failure and a (slowly) varying parameter.

Note that the same concept can be similarly applied on the sensor side of the plant. Consider

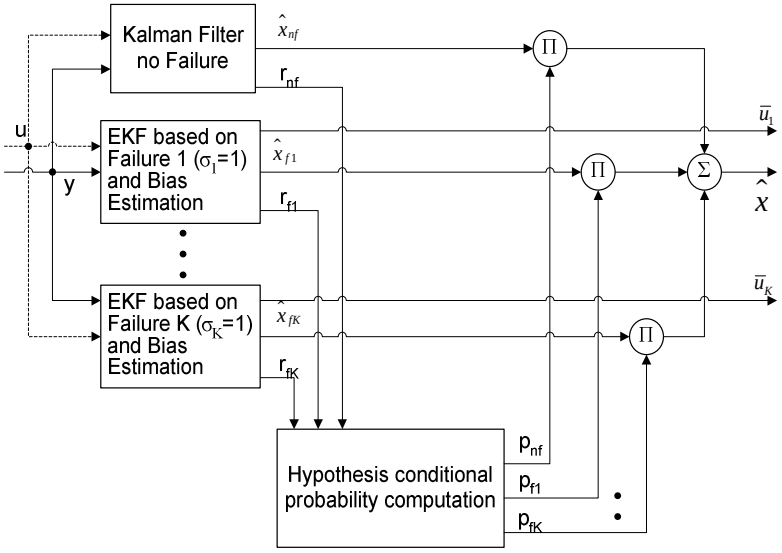

Figure 31: Scheme of the EMMAE method. The ordinary Kalman filters are replaced by EKF designed for state variable and actuator bias estimation.

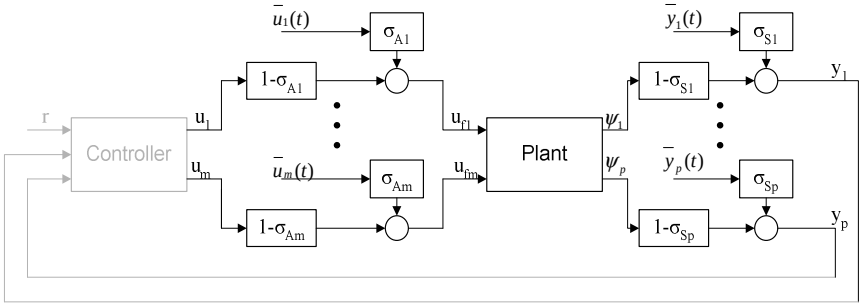

Figure 32: Control system with actuator and sensor failures

the control system with actuator and sensor failures shown in Figure 32. The sensor failures can be modeled as

$$y(t) = \psi(t) + \sigma_S(\bar{y}(t) - \psi(t)) \tag{4.4}$$

where $\psi(t)$ is the actual plant output and

$$\bar{y}(t) = \left[\bar{y}_1(t), \bar{y}_2(t), ..., \bar{y}_p(t)\right]^T, \sigma_S = diag\left\{\sigma_{S1}, \sigma_{S2}, ..., \sigma_{Sp}\right\} \tag{4.5}$$

with

$$\sigma_{Sj} = \begin{cases} 1 & \text{if the jth sensor fails} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

The next section provides a short introduction into the theory of parameter estimation with an EKF.

4.1 Parameter Estimation with an Extended Kalman Filter

Consider the nonlinear discrete system:

$$\begin{aligned} x(t+1) &= f(x(t),u(t),\theta(t)) + G_v(\theta)v(t) \\ x(0) &= \xi \\ y(t) &= h(x(t),u(t),\theta(t)) + r(t) \end{aligned} \tag{4.7}$$

where $v(t)$ and $r(t)$ are the system and measurement noises, and where their covariance matrices are

$$\begin{aligned} &E\left\{[v(t)][v(\tau)]^T\right\} = R_v(\theta)\delta_{t\tau} \quad \text{with } R_v(t,\theta) = R_v^T(\theta) \geq 0 \quad \text{for t},\tau \geq 0 \\ &E\left\{[r(t)][r(\tau)]^T\right\} = R_r(\theta)\delta_{t\tau} \quad \text{with } R_r(t,\theta) = R_r^T(\theta) > 0 \quad \text{for t},\tau \geq 0 \\ &E\left\{[v(t)][r(\tau)]^T\right\} = R_{vr}(\theta)\delta_{t\tau} \end{aligned} \tag{4.8}$$

In order to use a Kalman filter to predict the state variables of the system as well as to estimate its parameters, we augment the state vector with the parameters to be estimated [GOO-84]:

$$z = \begin{bmatrix} x \\ \theta \end{bmatrix} \tag{4.9}$$

The augmented state vector leads to the following nonlinear state space equations

$$\begin{aligned} z(t+1) &= f_z(z(t),u(t)) + \bar{G}_v(t)v(t) \\ y(t) &= h(z(t),u(t)) + r(t) \end{aligned} \tag{4.10}$$

with

$$f_z(z(t),u(t)) = \begin{bmatrix} f(z(t),u(t)) \\ \theta(t) \end{bmatrix} \tag{4.11}$$

and

$$\bar{G}_v(t) = \begin{bmatrix} G_v \\ 0 \end{bmatrix} \tag{4.12}$$

which implies that the system noise is not acting on the parameters.

To apply the Kalman filter equations to the nonlinear system above, the system has to be linearized continuously around its current working point. Since the current working point is not known at present, we use the prediction based on the data of the preceding time step. Hence the linear system matrices at time t are

$$F_z(t) = \frac{\partial}{\partial z} f_z(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = \begin{bmatrix} F(\hat{\theta}(t),\hat{x}(t),u(t)) & M(\hat{\theta}(t),\hat{x}(t),u(t)) \\ 0 & I \end{bmatrix} \tag{4.13}$$

where

$$F(\hat{\theta}(t),\hat{x}(t),u(t)) = \frac{\partial}{\partial x}[\, f(x(t),u(t),\theta(t))]\Big|_{z(t)=\hat{z}(t|t-1)} \triangleq F(t) \tag{4.14}$$

and

$$M(\hat{\theta}(t),\hat{x}(t),u(t)) = \frac{\partial}{\partial \theta}[\, f(x(t),u(t),\theta(t))]\Big|_{z(t)=\hat{z}(t|t-1)} \triangleq M(t) \tag{4.15}$$

The lower-right unity matrix of (4.13) implies that $\theta(t+1)=\theta(t)$. The linearized measurement matrix reads

$$\begin{aligned} &H_z(t) = \frac{\partial}{\partial z} h(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = [C_x(t) \quad C_\theta(t)] \\ &where: \\ &C_x(t) = \frac{\partial}{\partial x} h(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} \\ &C_\theta(t) = \frac{\partial}{\partial \theta} h(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} \end{aligned} \tag{4.16}$$

Now the Kalman predictor can be applied

Data Update:

$$\begin{aligned}\hat{z}(t|t) &= \hat{z}(t|t-1) + L_z(t)[y(t) - h(\hat{z}(t|t-1), u(t))] \\ \hat{z}(0|-1) &= \bar{\xi}_z\end{aligned} \tag{4.17}$$

$$\begin{aligned}\Sigma_z(t|t) &= [I - L_z(t)H_z(t)]\Sigma_z(t|t-1)[I - L_z(t)H_z(t)]^T + L_z(t)R_r(t)L_z^T(t) \\ \Sigma_z(0|-1) &= \Sigma_{z,0}\end{aligned} \tag{4.18}$$

$$L_z(t) = \Sigma_z(t|t-1)H_z^T(t)Q_z^{-1}(t) \tag{4.19}$$

$$Q_z(t) = H_z(t)\Sigma_z(t|t-1)H_z^T(t) + R_r(t) \tag{4.20}$$

Prediction:

$$\hat{z}(t+1|t) = f_z(\hat{z}(t|t-1), u(t)) + \bar{G}_v(t)R_{vr}(t)Q_z^{-1}(t)[y(t) - h(\hat{z}(t|t))] \tag{4.21}$$

$$\begin{aligned}\Sigma_z(t+1|t) = {} & F_z(t)\Sigma_z(t|t)F_z^T(t) + \bar{G}_v(t)R_v(t)\bar{G}_v^T(t) \\ & -\bar{G}_v(t)R_{vr}(t)Q_z^{-1}(t)R_{vr}^T(t)\bar{G}_v^T(t) \\ & -F_z(t)L_z(t)R_{vr}(t)\bar{G}_v^T(t) \\ & -\bar{G}_v(t)R_{vr}(t)L_z^T(t)F_z^T(t)\end{aligned} \tag{4.22}$$

Next $L(t,\theta)$ and $\Sigma(t|t)$ in equations (4.18) and (4.19) are partitioned according to the dimensions of $x(t)$ and $\theta(t)$

$$L_z(t) = \begin{bmatrix} L_x(t) \\ L_\theta(t) \end{bmatrix} \tag{4.23}$$

$$\Sigma_z(t|t) = \begin{bmatrix} \Sigma_1(t|t) & \Sigma_2(t|t) \\ \Sigma_2^T(t|t) & \Sigma_3(t|t) \end{bmatrix} \tag{4.24}$$

The decomposed filter equations now read

Data Update:

$$\begin{aligned}\hat{x}(t|t) &= \hat{x}(t|t-1) + L_x(t)[y(t) - h(\hat{z}(t|t-1), u(t))] \\ \hat{\theta}(t|t) &= \hat{\theta}(t|t-1) + L_\theta(t)[y(t) - h(\hat{z}(t|t-1), u(t))] \\ \hat{z}(0|-1) &= \overline{\xi}_z\end{aligned} \tag{4.25}$$

$$\begin{aligned}Q(t) = {} & C_x(t)\Sigma_1(t|t-1)C_x^T(t) + C_x(t)\Sigma_2(t|t-1)C_\theta^T(t) + C_\theta(t)\Sigma_2^T(t|t-1)C_x^T(t) \\ & + C_\theta(t)\Sigma_3(t|t-1)C_\theta^T(t) + R_r(t)\end{aligned} \tag{4.26}$$

$$\begin{aligned}L_x(t) &= [\Sigma_1(t|t-1)C_x^T(t) + \Sigma_2(t|t-1)C_\theta^T(t)]Q^{-1}(t) \\ L_\theta(t) &= [\Sigma_2^T(t|t-1)C_x^T(t) + \Sigma_3(t|t-1)C_\theta^T(t)]Q^{-1}(t)\end{aligned} \tag{4.27}$$

$$\begin{aligned}\Sigma_1(t|t) = {} & [I - L_x(t)C_x(t)]\Sigma_1(t|t-1)[I - L_x(t)C_x(t)]^T + L_x(t)C_\theta(t)\Sigma_3(t|t-1)C_\theta^T(t)L_x^T(t) \\ & - L_x(t)C_\theta(t)\Sigma_2^T(t|t-1)[I - L_x(t)C_x(t)]^T \\ & - [L_x(t)C_\theta(t)\Sigma_2^T(t|t-1)[I - L_x(t)C_x(t)]^T]^T + L_x(t)R_r(t)L_x^T(t)\end{aligned} \tag{4.28}$$

$$\begin{aligned}\Sigma_2(t|t) = {} & -[[I - L_x(t)C_x(t)]\Sigma_1(t|t-1) - L_x(t)C_\theta(t)\Sigma_2^T(t|t-1)]C_x^T(t)L_\theta^T(t) \\ & + [[I - L_x(t)C_x(t)]\Sigma_2(t|t-1) - L_x(t)C_\theta(t)\Sigma_3(t|t-1)][I - L_\theta(t)C_\theta(t)]^T \\ & + L_x(t)R_r(t)L_\theta^T(t)\end{aligned} \tag{4.29}$$

$$\begin{aligned}\Sigma_3(t|t) = {} & L_\theta(t)C_x(t)\Sigma_1(t|t-1)C_x^T(t)L_\theta^T(t) + [I - L_\theta(t)C_\theta(t)]\Sigma_3(t|t-1)[I - L_\theta(t)C_\theta(t)]^T \\ & - L_\theta(t)C_x(t)\Sigma_2(t|t-1)[I - L_\theta(t)C_\theta(t)]^T \\ & - [L_\theta(t)C_x(t)\Sigma_2(t|t-1)[I - L_\theta(t)C_\theta(t)]^T]^T \\ & + L_\theta(t)R_r(t)L_\theta^T(t)\end{aligned} \tag{4.30}$$

$$\Sigma(0|-1) = \Sigma_0 \tag{4.31}$$

Prediction:

$$\begin{aligned}\hat{x}(t+1|t) &= f(\hat{x}(t|t-1), u(t)) + G_v(t)R_{vr}(t)Q^{-1}(t)[y(t) - h(\hat{z}(t|t-1), u(t))] \\ \hat{\theta}(t+1|t) &= \hat{\theta}(t|t)\end{aligned} \tag{4.32}$$

$$\begin{aligned}\Sigma_1(t+1|t) = {} & F(t)\Sigma_1(t|t)F^T(t) + F(t)\Sigma_2(t|t)M^T(t) + M(t)\Sigma_2^T(t|t)F^T(t) \\ & + M(t)\Sigma_3(t|t)M^T(t) + G_v(t)R_v(t)G_v^T(t)\end{aligned} \tag{4.33}$$

$$\Sigma_2(t+1|t) = F(t)\Sigma_2(t|t) + M(t)\Sigma_3(t|t) \tag{4.34}$$

$$\Sigma_3(t+1|t) = \Sigma_3(t|t) \tag{4.35}$$

4.2 Failure-Parameter Estimation

In the following sections the actual EKF equations are derived for the case of failures of actuators or sensors. For the following sections we assume that the healthy plant (without failure parameters) can be described by the linear discrete-time system

$$\begin{aligned} x(t+1) &= Fx(t) + G_u u(t) + G_v v(t) \\ x(0) &= \xi \\ y(t) &= Hx(t) + r(t) \end{aligned} \tag{4.36}$$

This plant, together with the failure structure in Figure 32, can be described by the "failure" parameters σ_A and $\bar{u}$ from (4.2) for the actuator failures and σ_S and $\bar{y}$ from (4.5) for the sensor failures as follows

$$\begin{aligned} x(t+1) &= f(x(t), u(t), \sigma_A(t), \bar{u}(t)) + G_v v(t) \\ y(t) &= h(x(t), \sigma_S(t), \bar{y}(t)) + r(t) \end{aligned} \tag{4.37}$$

4.2.1 Actuator Failures

If we only look at actuator failures, we omit the sensor parameters σ_S and $\bar{y}$; hence with the system (4.36) the term $f(x(t), u(t), \sigma_A(t), \bar{u}(t))$ can be written as

$$f(x(t), u(t), \sigma_A(t), \bar{u}(t)) = Fx(t) + G_u \left[I_m - \sigma_A(t) \right] u(t) + G_u \sigma_A(t) \bar{u}(t) \tag{4.38}$$

For the derivation of an EKF for a failure hypothesis of the ith actuator we define $\sigma_{Ai} = 1$ and $\sigma_{Aj} = 0, \bar{u}_j = 0, \; j \neq i$. In order to apply an EKF, the state vector is augmented by the ith bias parameter.

$$z = \begin{bmatrix} x \\ \bar{u}_i \end{bmatrix} \tag{4.39}$$

The augmented state vector leads to the following nonlinear state space equations

$$\begin{aligned} z(t+1) &= f_z(z(t), u(t)) + \bar{G}_v(t) v(t) \\ y(t) &= h(z(t), u(t)) + r(t) \end{aligned} \tag{4.40}$$

with

$$f_z(z(t),u(t)) = \begin{bmatrix} f(z(t),u(t)) \\ \bar{u}_i(t) \end{bmatrix} \tag{4.41}$$

and

$$\bar{G}_v(t) = \begin{bmatrix} G_v \\ 0 \end{bmatrix} \tag{4.42}$$

Now the linearization of the dynamic matrix yields

$$F_z(t) = \frac{\partial}{\partial z} f_z(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = \begin{bmatrix} F & M(\hat{\bar{u}}_i(t),\hat{x}(t),u(t)) \\ 0 & 1 \end{bmatrix} \tag{4.43}$$

where

$$\begin{aligned} M(\hat{\bar{u}}_i(t),\hat{x}(t),u(t)) &= \frac{\partial}{\partial \bar{u}_i}\left[f(z(t),u(t))\right]\Big|_{z(t)=\hat{z}(t|t-1)} \\ &= G_u^{(i)} \end{aligned} \tag{4.44}$$

with $G_u^{(i)}$ representing the ith column of G_u. The input matrix becomes

$$G_z(t) = \frac{\partial}{\partial u} f_z(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = \begin{bmatrix} {}^{(0,i)}G_u \\ 0 \end{bmatrix} \tag{4.45}$$

with ${}^{(0,i)}G_u$ representing the matrix G_u with the ith column set to zero. The linearization of the measurement matrix is

$$\begin{aligned} &H_z(t) = \frac{\partial}{\partial z} h(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = [C_x(t) \quad C_\theta(t)] \\ &\textit{where}: \\ &C_x(t) = \frac{\partial}{\partial x} h(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = H \\ &C_\theta(t) = \frac{\partial}{\partial \bar{u}_i} h(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = 0 \end{aligned} \tag{4.46}$$

Now the linearized system can be written as

$$\begin{aligned}\begin{bmatrix} x(t+1) \\ \bar{u}_i(t+1) \end{bmatrix} &= \begin{bmatrix} F & G_u^{(i)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ \bar{u}_i(t) \end{bmatrix} + \begin{bmatrix} {}^{(0,i)}G_u \\ 0 \end{bmatrix} u(t) \\ y(t) &= [H \quad 0] \begin{bmatrix} x(t) \\ \bar{u}_i(t) \end{bmatrix}\end{aligned} \tag{4.47}$$

4.2.2 Sensor Failures

In case of sensor failures we omit the actuator parameters $\sigma_A(t)$ and $\bar{u}(t)$. The output equation is

$$h(x(t), \sigma_S(t), \bar{y}(t)) = \left[I_p - \sigma_S(t) \right] Hx(t) + \sigma_S(t) H\bar{y}(t) \tag{4.48}$$

and hence the output may be written as

$$y(t) = \left[I_p - \sigma_S(t) \right] Hx(t) + \sigma_S(t) H\bar{y}(t) + r(t) \tag{4.49}$$

For the derivation of an EKF for a failure hypothesis of the i[th] sensor we define $\sigma_{Si} = 1$ and $\sigma_{Sj} = 0, \bar{y}_j = 0, \ j \neq i$. In order to apply an EKF, the state vector is augmented by the i[th] bias parameter.

$$z = \begin{bmatrix} x \\ \bar{y}_i \end{bmatrix} \tag{4.50}$$

The augmented state vector leads to the following nonlinear state space equations

$$\begin{aligned} z(t+1) &= f_z(z(t), u(t)) + \bar{G}_v(t) v(t) \\ y(t) &= h(z(t), u(t)) + r(t) \end{aligned} \tag{4.51}$$

with

$$f_z(z(t), u(t)) = \begin{bmatrix} f(z(t), u(t)) \\ \bar{y}_i(t) \end{bmatrix} \tag{4.52}$$

and

$$\bar{G}_v(t) = \begin{bmatrix} G_v \\ 0 \end{bmatrix} \tag{4.53}$$

Now the linearization of the dynamic matrix yields

$$F_z(t) = \frac{\partial}{\partial z} f_z(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = \begin{bmatrix} F & M(\hat{\bar{y}}_i(t),\hat{x}(t),u(t)) \\ 0 & I \end{bmatrix} \tag{4.54}$$

where

$$\begin{aligned} M(\hat{\bar{y}}_i(t),\hat{x}(t),u(t)) &= \frac{\partial}{\partial \bar{y}_i}\left[f(z(t),u(t))\right]\Big|_{z(t)=\hat{z}(t|t-1)} \\ &= 0 \end{aligned} \tag{4.55}$$

The input matrix becomes

$$G_z(t) = \frac{\partial}{\partial u} f_z(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = \begin{bmatrix} G_u \\ 0 \end{bmatrix} \tag{4.56}$$

The linearization of the measurement Matrix is

$$\begin{aligned} &H_z(t) = \frac{\partial}{\partial z} h(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = [C_x(t) \quad C_\theta(t)] \\ &where: \\ &C_x(t) = \frac{\partial}{\partial x} h(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = {}^{(i,0)}H \\ &C_\theta(t) = \frac{\partial}{\partial \bar{y}_i} h(z(t),u(t))\Big|_{z(t)=\hat{z}(t|t-1)} = [0 \quad \ldots \quad 0 \quad \underbrace{1}_{i^{th}entry} \quad 0 \quad \ldots \quad 0]^T \end{aligned} \tag{4.57}$$

with ${}^{(i,0)}H$ representing the matrix H with the ith row set to zero. Now the linearized system can be written as

$$\begin{aligned} &\begin{bmatrix} x(t+1) \\ \bar{y}_i(t+1) \end{bmatrix} = \begin{bmatrix} F & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x(t) \\ \bar{y}_i(t) \end{bmatrix} + \begin{bmatrix} G_u \\ 0 \end{bmatrix} u(t) \\ &y(t) = [{}^{(i,0)}H \;\vdots\; [0 \quad \ldots \quad 0 \quad \underbrace{1}_{i^{th}entry} \quad 0 \quad \ldots \quad 0]^T]\begin{bmatrix} x(t) \\ \bar{y}_i(t) \end{bmatrix} \end{aligned} \tag{4.58}$$

4.2.3 Performance Improvement of Parameter Estimation

Equations (4.11) and (4.12) show that there is no noise acting directly on the parameter part of the augmented state vector. Thus the covariance matrix of the parameter error Σ_3 converges to a small value (norm) with increasing time. However, if a failure occurs the small value of Σ_3 prevents the parameter estimation from converging quickly to its new value. The parameter convergence rate can be improved by modeling the parameter as a pseudo-noise process: $\theta(t+1) = \theta(t) + v_p(t)$ [HIB-91], where $v_p(t)$ is a zero mean white noise process with mean square intensity Q_p; hence equation (4.10) can be rewritten as

$$z(t+1) = f_z(z(t),u(t)) + \bar{G}_v(t)v(t) + \begin{bmatrix} 0 \\ v_p(t) \end{bmatrix} \tag{4.59}$$

In order to include this pseudo-noise in the EKF equations, only the update step of the parameter error covariance matrix (4.30) needs to be modified as following

$$\begin{aligned} \Sigma_3(t|t) = {} & L_\theta(t)C_x(t)\Sigma_1(t|t-1)C_x^T(t)L_\theta^T(t) + [I - L_\theta(t)C_\theta(t)]\Sigma_3(t|t-1)[I - L_\theta(t)C_\theta(t)]^T \\ & - L_\theta(t)C_x(t)\Sigma_2(t|t-1)[I - L_\theta(t)C_\theta(t)]^T \\ & - [L_\theta(t)C_x(t)\Sigma_2(t|t-1)[I - L_\theta(t)C_\theta(t)]^T]^T \\ & + \mathrm{L}_\theta(t)R_r(t)L_\theta^T(t) + Q_p \end{aligned} \tag{4.60}$$

Now with the magnitude of Q_p the "liveliness" of the parameter estimation can be influenced in that a "large" value of Q_p leads to a lively, but somewhat inaccurate estimation, whereas a "small" value of Q_p reduces the parameter convergence rate.

4.3 Simulation Results

The failure detection with the EMMAE method was tested on the same nonlinear model as the MMAE method described in Section 3.5. Three actuators and one sensor were "equipped" with an EKF (ailerons 1, 2, rudder 1, and roll rate sensor 1). The parameters for the (extended) Kalman filters are chosen as

$$\Sigma_{10} = 1I_n,\ \Sigma_{30} = 0.1,\ R_v = 0.001I_m,\ R_r = 1I_p,\ Q_p = 0.001 \tag{4.61}$$

In a first experiment a lock-in-place failure of the second aileron at 4.5 s is investigated. Figure 33 shows that the second aileron is stuck at the peak of the input amplitude. The actuator bias is estimated by the EKF. Figure 34 shows that the detection of the failure is delayed by about 1 s. In a further simulation we investigate the same scenario, but this time the actuator is not just stuck in place but "jumps" to another position (see Figure 35). Figure 36 shows that this time the failure is detected immediately. However, during the first two seconds after the failure there is an ambiguity with the failure hypothesis of aileron 1. The immediate detection may be due to the excitation from the jump of the actuator position.

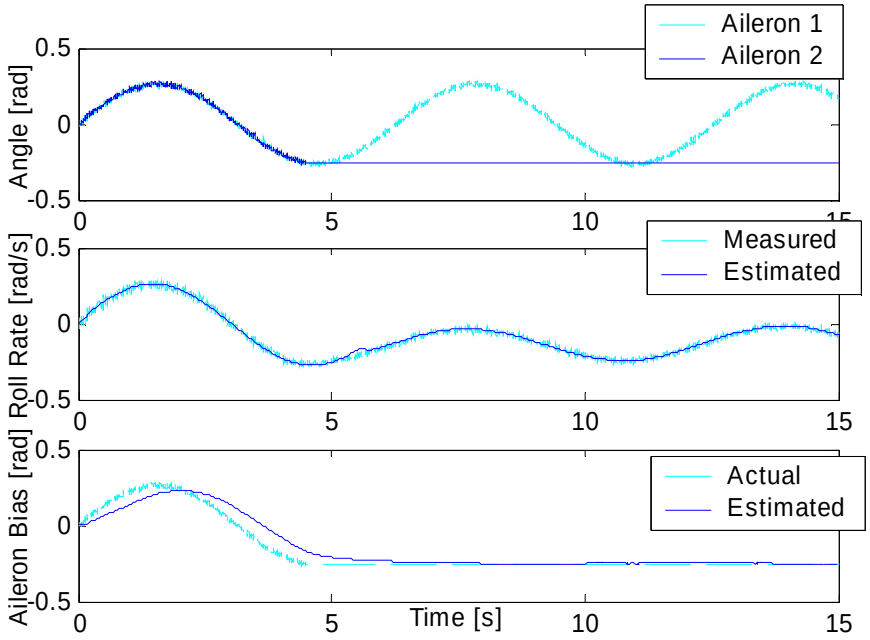

Figure 33: Aileron input signal and roll rate. Lock-in-place failure of aileron 2 at 4.5 s. The aileron bias is estimated by the EKF. Inputs: $u_1(t)=u_2(t)=0.2618\ \sin(t)$, $u_3(t)=u_6(t)=0$, $u_4(t)=u_5(t)=0.2618\sin(t+\pi)$. Efficiency coefficients of ailerons: $c_1=1$, $c_2=0.6$.

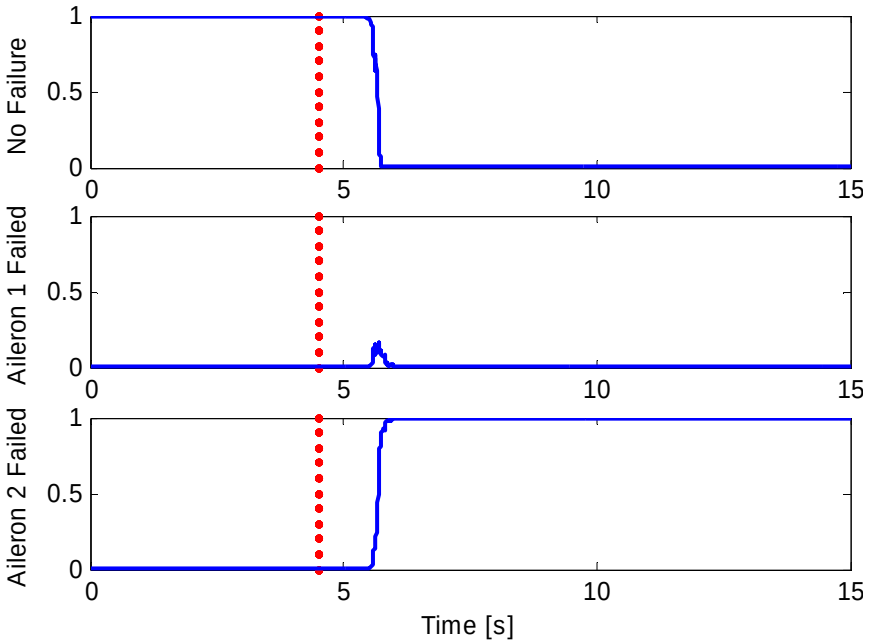

Figure 34: Probability that either no failure or a failure of aileron 1 or 2 has occurred. The dots indicate the time of the failure. With a delay of about 1 s the failure is detected and isolated.

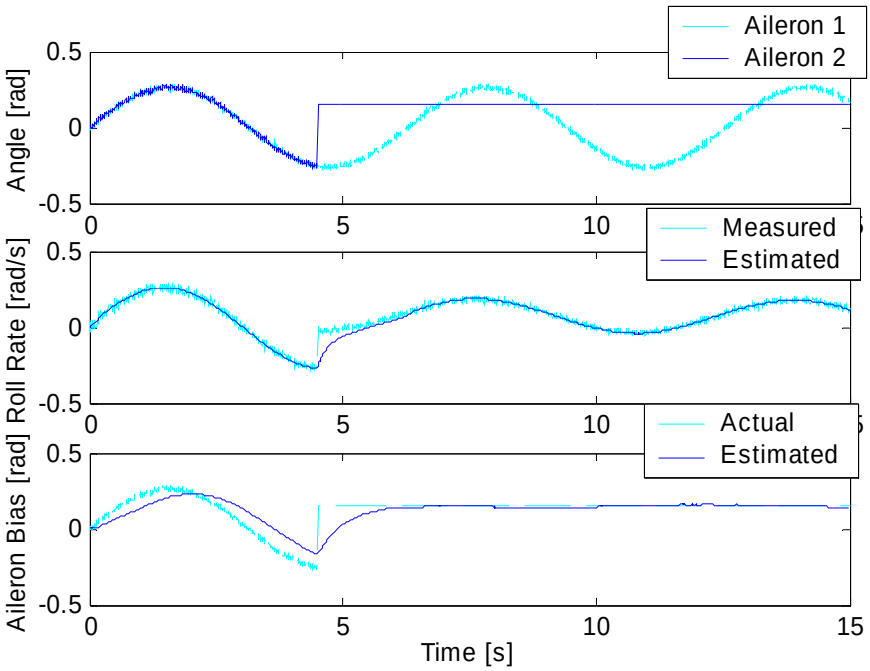

Figure 35: Aileron input signal and roll rate. Excursive lock-in-place failure of aileron 2 at 4.5 s. The. Inputs: $u_1(t)=u_2(t)=0.2618\ \sin(t)$, $u_3(t)=u_6(t)=0$, $u_4(t)=u_5(t)=0.2618\sin(t+\pi)$. Efficiency coefficients of ailerons: $c_1=1$, $c_2=0.6$.

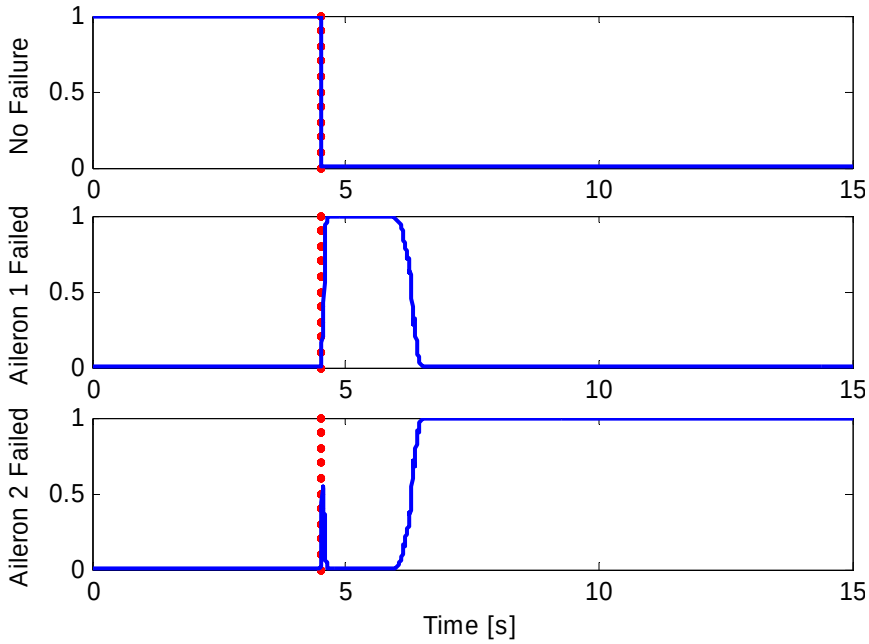

Figure 36: Probability that either no failure or a failure of aileron 1 or 2 has occurred. The dots indicate the time of the failure. The jump in the angle of aileron leads to an immediate failure detection, however, the isolation is delayed due to confusion with aileron 1.

As mentioned above the EMMAE algorithm can also cope with varying failures. In the following simulation we assume that the second aileron fails after 5 s but then still moves in the manner of a square. Figure 37 shows that the roll rate is estimated quite well despite the varying failure. In order to accelerate the estimation of the bias parameter, the mean square intensity Q_p of the pseudo-noise is increased to $Q_p = 0.0005$. Figure 38 shows that the failure is detected immediately, however, during the first 4.5 s after the failure there is an ambiguity with the failure hypothesis of the first aileron. The reason for this may (again) be the larger coefficient of efficiency of aileron 1.

In the last simulations of this chapter a varying sensor failure scenario is assumed. After 4.5 s the first roll rate sensor emits a sine signal that is not correlated to the actual measurement (Figure 39). The sine function was chosen as $h_f(t) = 0.5\sin(2t) + 0.5$ and in order to get an accurate estimation of the failure signal the square intensity Q_p of the pseudo-noise is increased to $Q_p = 0.01$. Figure 40 shows an immediate detection and isolation of the sensor failure.

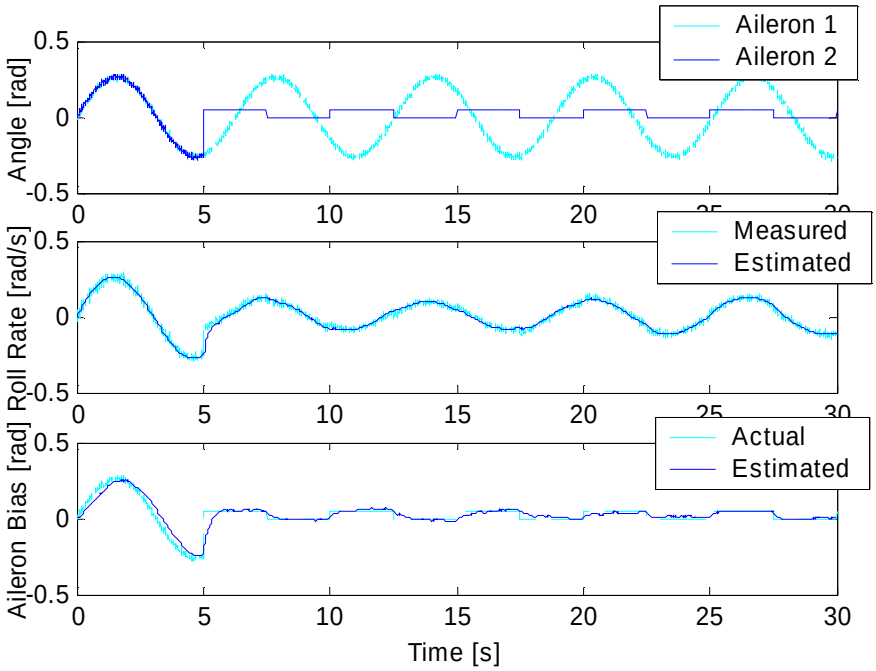

Figure 37: Aileron input signal and roll rate. Varying lock-in-place failure of aileron 1 starting at 5 s. The inputs: $u_1(t)=u_2(t)=0.2618\ \sin(t)$, $u_3(t)=u_6(t)=0$, $u_4(t)=u_5(t)=0.2618\sin(t+\pi)$. Efficiency coefficients of ailerons: $c_1=1$, $c_2=0.6$. The mean square intensity of the pseudo noise is 0.0005.

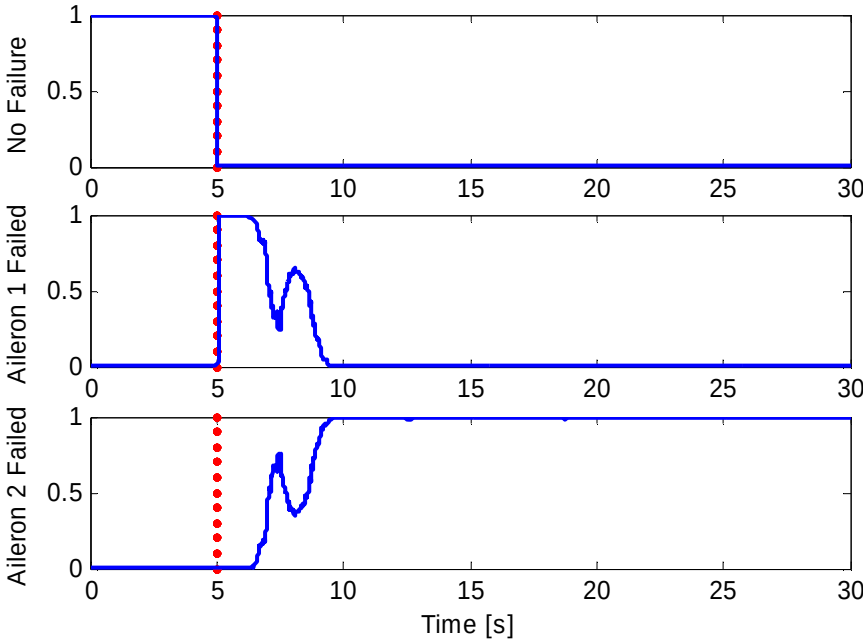

Figure 38: Probability that either no failure or a failure of aileron 1 or 2 has occurred. The dots indicate the time of the failure. The failure is detected immediately, but isolated only after about 4.5 s.

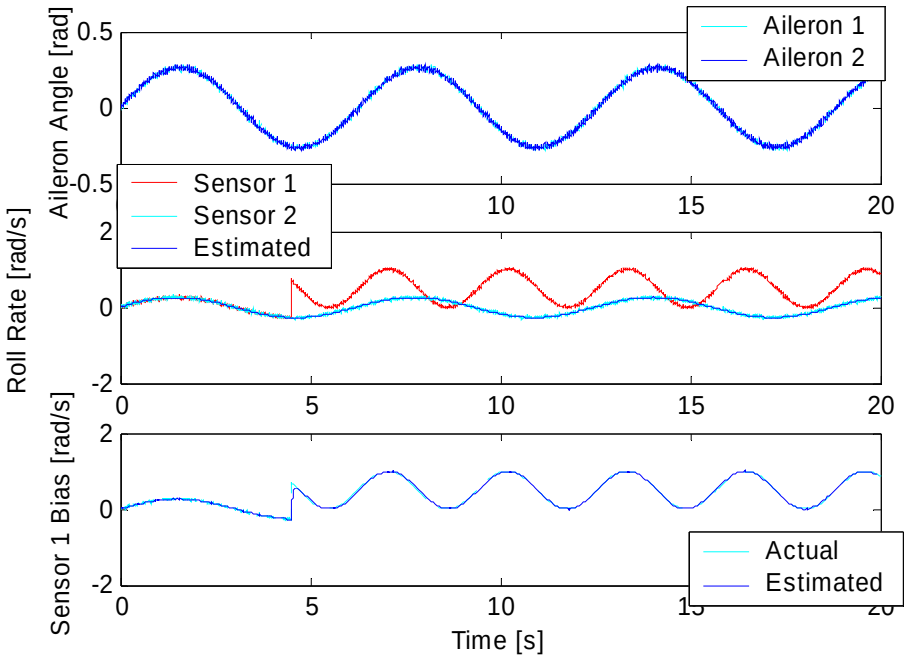

Figure 39: Aileron input signal and roll rate. Varying lock-in-place failure of roll rate sensor 1 starting at 4.5 s; Sensor failure signal: hf(t)=0.5+0.5sin(2t). The inputs: $u_1(t)=u_2(t)=0.2618\ \sin(t)$, $u_3(t)=u_6(t)=0$, $u_4(t)=u_5(t)=\ 0.2618\sin(t+\pi)$. Efficiency coefficients of ailerons: $c_1=1$, $c_2=0.6$. The mean square intensity of the pseudo noise is 0.01.

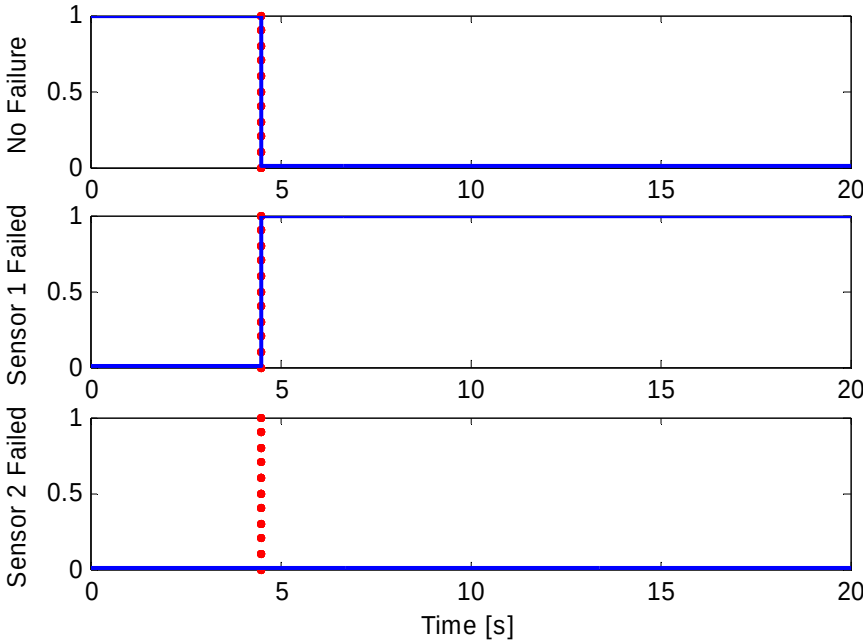

Figure 40: Probability that either no failure or a failure of roll rate sensor 1 or 2 has occurred. The dots indicate the time of the failure. The failure is detected and isolated immediately.

5 Nonlinear Aircraft Model

In this chapter a nonlinear dynamic model of a model aircraft is presented. However, the resulting model is only needed for benchmark purposes and therefore the modeling process itself is presented rather briefly. Furthermore, the parameters used for the simulation have not been identified with data from a real aircraft, but represent reasonable guesses only. For simulation purposes the model is implemented as an S-function in MATLAB/Simulink.

5.1 Modeling of the Aircraft

The resulting tenth-order model has the following state variables

$$x(t)=\begin{bmatrix} \mathrm{x}_1(t) \\ \mathrm{x}_2(t) \\ \mathrm{x}_3(t) \\ \mathrm{x}_4(t) \\ \mathrm{x}_5(t) \\ \mathrm{x}_6(t) \\ \mathrm{x}_7(t) \\ \mathrm{x}_8(t) \\ \mathrm{x}_9(t) \\ \mathrm{x}_{10}(t) \end{bmatrix} \triangleq \begin{bmatrix} \mathrm{p}(t) \\ q(t) \\ r(t) \\ q_0(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ u(t) \\ v(t) \\ w(t) \end{bmatrix} \begin{array}{l} \text{roll rate } [\text{rad/s}] \\ \text{nick rate } [\text{rad/s}] \\ \text{yaw rate } [\text{rad/s}] \\ \text{Euler parameter } [\text{-}] \\ \text{"} \\ \text{"} \\ \text{"} \\ \text{longitudinal velocity } [\text{m/s}] \\ \text{lateral velocity } [\text{m/s}] \\ \text{normal velocity } [\text{m/s}] \end{array} \tag{5.1}$$

The inputs are

$$u(t)=\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix} \triangleq \begin{bmatrix} \xi(t) \\ \eta(t) \\ \zeta(t) \\ \alpha_{th}(t) \end{bmatrix} \begin{array}{l} \text{aileron angle [rad]} \\ \text{elevator angle [rad]} \\ \text{rudder angle [rad]} \\ \text{throttle angle [rad]} \end{array} \tag{5.2}$$

The outputs are given by the (linear) output equations

$$y(t)=\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \\ y_5(t) \\ y_6(t) \end{bmatrix} \triangleq \begin{bmatrix} p(t) \\ q(t) \\ r(t) \\ u(t) \\ v(t) \\ w(t) \end{bmatrix} \begin{array}{l} \text{roll rate } [\text{rad/s}] \\ \text{nick rate } [\text{rad/s}] \\ \text{yaw rate } [\text{rad/s}] \\ \text{longitudinal velocity } [\text{m/s}] \\ \text{lateral velocity } [\text{m/s}] \\ \text{normal velocity } [\text{m/s}] \end{array} \tag{5.3}$$

In practice, the velocities in the body frame cannot be measured directly. However, for our theoretical investigations the velocities are assumed to be measurable in order to get an observable system.

The aerodynamic forces acting on the aircraft are specified in the wind tunnel coordinate system

$$ {}_{wt}F(t) = \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix}_{wt} = \begin{bmatrix} 1/2\rho V^2 S C_x(\alpha,\beta) \\ 1/2\rho V^2 S C_y(\alpha,\beta) \\ 1/2\rho V^2 S C_z(\alpha) \end{bmatrix} \tag{5.4} $$

with

X : drag force [N]	C_z: lift coefficient [-]
Y : lateral force [N]	C_y: lateral coefficient [-]
Z : lift force [N]	$V = \sqrt{u^2+v^2+w^2}$: wind velocity [m/s]
S : projected surface [m^2]	α = arctan(w/u) : incidence angle [rad]
ρ : air density [kg/m^3]	β = arcsin(v/V) : sideslip angle [rad]
C_x: drag coefficient [-]	

The weight force of the plane in the navigation frame is

$$ {}_{n}F_w(t) = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{5.5} $$

where m is the overall aircraft mass. The thrust force in the body frame is

$$ {}_{b}F_t(t) = \begin{bmatrix} C_\alpha \alpha_{th} \\ 0 \\ 0 \end{bmatrix} \tag{5.6} $$

where C_α is a thrust factor. The equations of motion require all forces to be transformed into a body frame description

$$ {}_{b}F(t) = C_{bw}(\alpha,\beta)\,{}_{wt}F(t) + {}_{b}F_t(t) + C_{bn}(q_0,q_1,q_2,q_3)\,{}_{n}F_w(t) \tag{5.7} $$

where $C_{bw}(\alpha,\beta)$ is the transformation matrix from the wind tunnel to the body frame and $C_{bn}(q_0,q_1,q_2,q_3)$ is the transformation matrix from the navigation to the body frame.

The moments acting on the aircraft specified in body coordinates are

$$ {}_bM(t) = \begin{bmatrix} L(t) \\ M(t) \\ N(t) \end{bmatrix}_b = \begin{bmatrix} 1/2\rho V^2 SbC_L(\beta, p, r, \xi) \\ 1/2\rho V^2 Sl_\mu C_M(\alpha, q, \eta) \\ 1/2\rho V^2 SbC_N(\beta, r, \zeta) \end{bmatrix} \tag{5.8} $$

with

L : roll moment [Nm]	C_x: rolling moment coefficient [-]
M : pitch moment [Nm]	C_y: pitching moment coefficient [-]
N : yaw moment [Nm]	C_z: yawing moment coefficient [-]
l_μ : aerodynamic mean chord [m]	b : wingspan [m]

With these preliminaries the equations of motion can be derived. The equations of forces are

$$ \begin{bmatrix} \dot{u}(t) \\ \dot{v}(t) \\ \dot{w}(t) \end{bmatrix} = \frac{1}{m} {}_bF(t) + {}_bF(t) \times {}_b\omega(t) \tag{5.9} $$

with

$$ {}_b\omega(t) = \begin{bmatrix} p(t) \\ q(t) \\ r(t) \end{bmatrix} \tag{5.10} $$

The equations of moments are

$$ {}_b\dot{\omega}(t) = I^{-1}\left({}_bM(t) - {}_b\omega(t) \times \left(I\, {}_b\omega(t) \right) \right) \tag{5.11} $$

with I being the inertial tensor. The differential equation for the Euler parameters can be calculated according to the theory of quaternions [HIL-03]

$$ {}_n\dot{Q}(t) = \frac{1}{2} Q(t)\, {}_b\omega(t) \tag{5.12} $$

where the quaternion Q can be written as

$$Q = \begin{bmatrix} q_0(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \end{bmatrix} \tag{5.13}$$

The Euler parameters are normalized such that the norm of the quaternion $N(Q) = q_0^2 + \underline{q} \cdot \underline{q}$ equals one. Note that $\omega(t)$ can be seen as a quaternion with a real part of zero. Hence according to the multiplication rule of quaternions we obtain the differential equation in matrix form

$$\begin{bmatrix} \dot{q}_0(t) \\ \dot{q}_1(t) \\ \dot{q}_2(t) \\ \dot{q}_3(t) \end{bmatrix}_n = \frac{1}{2} \begin{bmatrix} -q_1(t) & -q_2(t) & -q_3(t) \\ q_0(t) & -q_3(t) & q_2(t) \\ q_3(t) & q_0(t) & -q_1(t) \\ -q_2(t) & q_1(t) & q_0(t) \end{bmatrix}_b \omega(t) \tag{5.14}$$

5.2 Addition of Redundancies

In order to control a system susceptible to sensor and actuator faults, appropriate redundancies need to be added. In general, the plant has to be so constructed that in case of any possible actuator or sensor failures the controller can still achieve a desired control objective. In the case of our nonlinear model the redundancies are added by "cloning" the required components. For the actuators in the nonlinear model this is done by substituting a given control signal by a sum of (redundant) control signals. For instance, if we want to implement redundant ailerons, the equation for the roll moments acting on the plane (5.8) needs to be modified as follows:

$$L(t) = 1/2\,\rho V^2 SbC_L(\beta, p, r, \xi_{red}) \tag{5.15}$$

where ξ_{red} represents a group of m ailerons

$$\xi_{red} = c_1\xi + c_2\xi + \ldots + c_m\xi \tag{5.16}$$

with the coefficient c_i indicating different efficiencies.

In the case of sensors, redundancies can be added by appending the desired measurement equation. In our case where the measurement equations are linear, a sensor can be "cloned" by inserting a new row into the measurement matrix.

5.3 SISO Model of Vertical Dynamics

In order to investigate the SISO MRAC algorithm, the nonlinear model above is reduced to the vertical dynamics. The state variables are

$$x(t)=\begin{bmatrix} \mathrm{x}_1(t) \\ \mathrm{x}_2(t) \\ \mathrm{x}_3(t) \\ \mathrm{x}_4(t) \end{bmatrix} \triangleq \begin{bmatrix} \theta(t) \\ q(t) \\ u(t) \\ w(t) \end{bmatrix} \begin{matrix} \text{pitch attitude } [\text{rad}] \\ \text{pitch rate } [\text{rad/s}] \\ \text{longitudinal velocity } [\text{m/s}] \\ \text{normal velocity } [\text{m/s}] \end{matrix} \tag{5.17}$$

The input is the elevator angle

$$u(t) \triangleq \eta(t) \tag{5.18}$$

and the output is the pitch attitude

$$y(t) \triangleq \theta(t) \tag{5.19}$$

The aerodynamic forces acting on the aircraft are specified in the wind tunnel coordinate system

$${}_{wt}F(t)=\begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix}_{wt}=\begin{bmatrix} 1/2\rho V^2 S C_x(\alpha) \\ 0 \\ 1/2\rho V^2 S C_z(\alpha) \end{bmatrix} \tag{5.20}$$

Note that all lateral forces are of no interest because we are only interested in the vertical dynamics. Since the pitch attitude is known, the weight force (5.5) of the plane can be easily transformed into body frame

$${}_bF_w(t)=\begin{bmatrix} -\mathrm{mg}\sin(\theta) \\ 0 \\ \mathrm{mg}\cos(\theta) \end{bmatrix} \tag{5.21}$$

The side-slip angle β is always zero. Hence the transformation of the aerodynamic forces from wind tunnel to navigation frame is

$$ {}_bF(t) = \begin{bmatrix} -X\cos(\alpha)+Z\sin(\alpha) \\ 0 \\ -X\sin(\alpha)-Z\cos(\alpha) \end{bmatrix} \tag{5.22} $$

For the thrust force we assume a constant value

$$ {}_bF_t(t) = \begin{bmatrix} F_t \\ 0 \\ 0 \end{bmatrix} \tag{5.23} $$

The total force acting on the airplane can thus be written as

$$ {}_bF(t) = {}_bF(t) + {}_bF_t(t) + {}_bF_w(t) \tag{5.24} $$

Since only the vertical dynamics are considered, the moments acting on the aircraft can be reduced to the pitching moment

$$ {}_bM(t) = {}_b\begin{bmatrix} L(t) \\ M(t) \\ N(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2\rho V^2 S l_\mu C_M(\alpha, q, \eta) \\ 0 \end{bmatrix} \tag{5.25} $$

Based on the definitions above, the equations of motion can be derived as in the previous section.

6 Conclusions

The MRAC method is a thorough technique for the design of a control strategy for systems with actuator failures. As soon as the conditions for the existence of a stable MRAC controller are met, the designer is given a versatile tool to develop a control system which has the desired properties. The simulations with the nonlinear model show some promising results. As long as the design is based on a carefully chosen model and as long as the number of parameters to be adapted is kept sufficiently small, the design process is straightforward. However, the conditions mentioned above may sometimes be quite restrictive. The necessity of a minimum-phase system can be cumbersome since a nonlinear system is often not minimum-phase, at least not over the whole operation area. In the literature on adaptive control systems nonminimum-phase systems are usually treated with a different control design method known as adaptive pole placement. In this work only SISO (redundant) systems are treated. The extension to a practicable MIMO theory is far from trivial and is a topic of current research.

The primary needs for a failure detection mechanism, more than a complete control design solution, drive the focus of the second part of this work. With the MMAE method, a practicable and powerful method for the detection of actuator as well as sensor failures is presented. The MMAE method is a good choice as long as the expected failures can be hypothesized by a reasonable number of Kalman filters. The simulations with a full-degree-of-freedom nonlinear model give auspicious results; however the results obtained here show that the kind of addressable failures is rather restricted. In order to release this restriction, the MMAE algorithm is combined with the parameter-estimating ability of an EKF. The reliable ability of the MMAE method to detect failures based on a predefined hypothesis and the fact that each EKF only has to estimate one single (failure-)parameter lead to a formation of a very fruitful synergy. The simulation results show the capability of the EMMAE method to detect failures of various kinds. However, both, the MMAE and the EMMAE method require further investigations, including ways to reduce the computational burden, control strategies and analysis of the in-the-loop behavior, as well as the treatment of multiple failures.

Appendix

Appendix A.1

Existence of a Solution of the Model-Matching Equation

In order to show the existence of a solution for the expression

$$\left(\Lambda(s)-\theta_1^{*T}a(s)\right)P(s)-\left(\theta_2^{*T}a(s)+\theta_0^{*}\Lambda(s)\right)k_pZ_a(s)=\Lambda(s)Z_a(s)P_m(s) \tag{A.1}$$

we introduce the Bezout Identity given in the following Lemma (without proof)

Lemma 2:

Let A(s) and B(s) be monic polynomials of degrees n and $m \le n-1$, respectively, which are relatively prime (no mutual zeros). Then, the polynomial Q(s) and the monic polynomial R(s) of degree n-1 exist such that

$$A(s)R(s)+B(s)Q(s)=C(s) \tag{A.2}$$

where C(s) is an arbitrary polynomial of degree 2n-1.

Now with

$$\begin{aligned} A(s) &= P(s) \\ B(s) &= Z_a(s) \\ R(s) &= \Lambda(s)-\theta_1^{*T}a(s) \\ Q(s) &= -k_p\left(\theta_2^{*T}a(s)+\theta_0^{*}\Lambda(s)\right) \end{aligned} \tag{A.3}$$

the (arbitrary) polynomial $C(s)=\Lambda(s)Z_a(s)P_m(s)$ can be constructed if the polynomials R(s) and Q(s) can be chosen arbitrarily. With the definition of a(s) and $\Lambda(s)$ in Section 2.2.1, they can be written as

$$\begin{aligned} R(s) &= s^{n-1}+\left(\lambda_{n-2}-\theta_{1,n-2}^{*}\right)s^{n-2}+\ldots+\left(\lambda_1-\theta_{1,1}^{*}\right)s^1+\left(\lambda_0-\theta_{1,0}^{*}\right) \\ Q(s) &= -k_p\left[\theta_0^{*}s^{n-1}+\left(\theta_0^{*}\lambda_{n-2}+\theta_{2,n-2}^{*}\right)s^{n-2}+\ldots+\left(\theta_0^{*}\lambda_1+\theta_{2,1}^{*}\right)s^1+\left(\theta_0^{*}\lambda_0+\theta_{2,0}^{*}\right)\right] \end{aligned} \tag{A.4}$$

and hence all desired polynomials can be constructed with the appropriate choice of the parameter vectors $\theta_0^{*},\theta_1^{*},\theta_2^{*}$.

Appendix A.2

Model Parameters

The physical parameters are

$$
\begin{array}{lcll}
m & = & 28 & [kg] \quad \text{Weight of the aircraft} \\
I & = & \begin{bmatrix} 2.24 & 0 & 0.5 \\ 0 & 6.37 & 0 \\ 0.5 & 0 & 6.37 \end{bmatrix} & [kgm^2] \quad \text{Inertial tensor} \\
l_{\mu} & = & 0.9 & [m] \quad \text{Aerodynamic mean chord} \\
S & = & 2.5 & [m^2] \quad \text{Projected surface} \\
g & = & 9.81 & [m/s^2] \quad \text{Acceleration of gravity} \\
\rho & = & 1.167 & [kg/m^3] \quad \text{Air density}
\end{array}
\tag{A.5}
$$

The aerodynamical coefficients are

$$
\begin{aligned}
\frac{dC_x}{d\beta} &= 0.1 && [\text{rad}^{-1}] \\
\frac{dC_y}{d\beta} &= -1.0 && [\text{rad}^{-1}] \\
C_{z0} &= 0 && [-] \\
\frac{dC_z}{d\alpha} &= 4.5 && [\text{rad}^{-1}] \\
\frac{dC_L}{d\beta} &= -0.1 && [\text{s rad}^{-1}] \\
\frac{dC_L}{dp} &= -0.1 && [-] \\
\frac{dC_L}{dr} &= 0.02 && [\text{s rad}^{-1}] \\
\frac{dC_L}{d\xi} &= 0.1 && [-] \\
C_{M0} &= 0 && [-] \\
\frac{dC_M}{d\alpha} &= -0.1 && [\text{rad}^{-1}] \\
\frac{dC_M}{dq} &= -0.1 && [\text{s rad}^{-1}] \\
\frac{dC_M}{d\eta} &= 0.162 && [-] \\
\frac{dC_N}{d\beta} &= 0.1 && [\text{rad}^{-1}] \\
\frac{dC_N}{dr} &= -0.1 && [\text{s rad}^{-1}] \\
\frac{dC_N}{d\zeta} &= 0.1 && [-]
\end{aligned}
\qquad \text{(A.6)}
$$

Appendix A.3

Linearized System

With the parameters from Appendix A.2 the system matrices of Section 3.5 are

$$A = \begin{bmatrix} -88.0302 & 0 & 24.5158 & 0 & 0 & 0 & 0 & 0 & -2.47149 & 0 \\ 0 & -30.4133 & 0 & 0 & 0 & 0 & 0 & 0.0223235 & 0 & -0.791409 \\ 6.90975 & 0 & -32.3376 & 0 & 0 & 0 & 0 & 0 & 0.985719 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.08312 & 0 & 0 & 0 & -19.62 & 0 & -0.0933396 & 0 & 0.0117212 \\ 1.08312 & 0 & -38.3988 & 0 & 19.62 & 0 & 0 & 0 & -2.05497 & 0 \\ 0 & 38.3988 & 0 & 0 & 0 & 0 & 0 & -0.255015 & 0 & -9.07353 \end{bmatrix}$$

$$B = \begin{bmatrix} 88.0302 & 52.8181 & 0 & -6.90975 & -4.14585 & 0 \\ 0 & 0 & 49.2696 & 0 & 0 & 0 \\ -6.90975 & -4.14585 & 0 & 30.9557 & 18.5734 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.03571 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.7}$$

The dynamic matrix shows that the fourth and the seventh state variable have no influence on the input/output behavior. Therefore, for our purpose they may be dropped, leading to the new system matrices

$$A = \begin{bmatrix} -88.0302 & 0 & 24.5158 & 0 & 0 & 0 & -2.47149 & 0 \\ 0 & -30.4133 & 0 & 0 & 0 & 0.0223235 & 0 & -0.791409 \\ 6.90975 & 0 & -32.3376 & 0 & 0 & 0 & 0.985719 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.08312 & 0 & 0 & -19.62 & -0.0933396 & 0 & 0.0117212 \\ 1.08312 & 0 & -38.3988 & 19.62 & 0 & 0 & -2.05497 & 0 \\ 0 & 38.3988 & 0 & 0 & 0 & -0.255015 & 0 & -9.07353 \end{bmatrix}$$

$$B = \begin{bmatrix} 88.0302 & 52.8181 & 0 & -6.90975 & -4.14585 & 0 \\ 0 & 0 & 49.2696 & 0 & 0 & 0 \\ -6.90975 & -4.14585 & 0 & 30.9557 & 18.5734 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.03571 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.8}$$

The normalization matrices have been chosen as

$$T_x = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.03 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_u = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_y = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A.9)}$$

Appendix A.4

Equation for Forces and Moments of the Nonlinear Model

The forces in the wind tunnel coordinate system are modeled as

$$\begin{aligned} X &= \frac{1}{2}\rho V^2 s\left(43.6\alpha^4 - 8.47\alpha^3 + 1.33\alpha^2 + 0.1887\alpha + 0.0206 + \frac{dC_x}{d\beta}|\beta|; \right) \\ Y &= \frac{1}{2}\rho V^2 s\left(\frac{dC_y}{d\beta}\beta \right) \\ Z &= \frac{1}{2}\rho V^2 s\left(C_{z0} + \frac{dC_z}{d\alpha}\alpha \right) \end{aligned} \tag{A.10}$$

and the moments in body frame:

$$\begin{aligned} L &= \frac{1}{2}\rho V^2 sb\left(\beta\frac{dC_L}{d\beta} + p\frac{dC_L}{dp} + r\frac{dC_L}{dr} + \xi\frac{dC_L}{d\xi} \right) \\ M &= \frac{1}{2}\rho V^2 sl_\mu\left(dC_{M0} + \alpha\frac{dC_M}{d\alpha} + q\frac{dC_M}{dq} + \eta\frac{dC_M}{d\eta} \right) \\ N &= \frac{1}{2}\rho V^2 sb\left(\beta\frac{dC_N}{d\beta} + r\frac{dC_N}{dr} + \zeta\frac{dC_N}{d\zeta} \right) \end{aligned} \tag{A.11}$$

with aircraft velocity V.

Appendix A.5
Symbols and Abbreviations

Symbols – Chapter 2

Symbol	Description
A,B,C	State space system matrices
u(t)	Input vector
x(t)	State vector
y(t)	Ouput vector
$\bar{u}(t)$	Input bias
$v(t)$	Controller output
σ	Failure variable
$Z_j(s)$	Numerator polynomial of jth transfer function
$P(s)$	Denominator polynomial of transfer function
k_{pj}	High-frequency gain of jth transfer function
$G(s)$	Transfer function
$G_m(s)$	Reference transfer function
$P_m(s)$	Reference denominator polynomial
$y_m(t)$	Reference output
k_m	Reference high-frequency gain
n^*	Relative degree of transfer function
$\bar{Z}_m$	Reference numerator polynomial without prefilter
$\bar{P}_m$	Reference denominator polynomial without prefilter
$\bar{r}(t)$	Unfiltered reference signal
$r(t)$	Filtered reference signal
$v_0(t)$	Common controller ouput
$\theta_1, \theta_2, \theta_0, k, \theta_4$	Control parameters
$\theta_1^*, \theta_2^*, \theta_0^*, k^*, \theta_4^*$	Matching control parameters
$\omega_1(t), \omega_2(t)$	Auxiliary signals

Symbol	Description
$a(s)$	Numerator vector of auxiliary signal transfer function
$\Lambda(s)$	Denominator of auxiliary signal transfer function
$\bar{y}(t)$	Output due to failure
$\phi(t)$	Parameter error
$\omega(t)$	Control signal vector
$\theta(t)$	Control parameter vector
$e(t)$	Matching error
$\varepsilon(t)$	Extended matching error
$\upsilon(t)$	Additional error signal
$L(s)$	Auxiliary transfer function
$k_1(t)$	Estimated high-frequency gain
$\rho(t)$	Estimation error of high-frequency gain
u_{nom}, x_{nom}	Nominal values
u_{red}	Redundant input vector
$\bar{W}_c, \bar{W}_o$	Joint grammians
Γ, γ	Adaptation gains
s_i	Ith pole of transfer function

Symbols – Chapter 3

Symbol	Description
$\hat{x}_{nf}$	State vector estimation of Kalman filter based on no failure
r_{nf}	Residual of Kalman filter based on no failure
p_{nf}	Probability of no-failure hypothesis
$\hat{x}_{if}$	State vector estimation of Kalman filter based on ith failure
r_{if}	Residual of Kalman filter based on ith failure
p_{if}	Probability of ith failure hypothesis
θ	Failure parameter

Symbol	Description
$F(\theta)$	Discrete dynamic matrix
$G_u(\theta)$	Discrete input matrix
$G_v(\theta)$	Input-noise gain matrix
$H(\theta)$	Discrete measurement matrix
$v(t)$	System noise
$r(t)$	Measurement noise
ξ	Initial values of state vector
$\bar{\xi}$	Expectation of initial value of state vector
K	Kalman gain
$R_v(\theta)$	Covariance matrix of system noise
$R_r(\theta)$	Covariance matrix of measurement noise
$R_{vr}(\theta)$	Cross-covariance matrix of system and measurement noise
$Q(\theta)$	Residual covariance matrix
$L(t)$	Kalman update gain
$\Sigma_0(t)$	Initial covariance matrix of state prediction error
$\Sigma(t)$	Covariance matrix of state prediction error
$\Sigma_{subopt}(t)$	Suboptimal covariance matrix of state prediction error
$e(t)$	Estimation error
t_s	Sample time
I_n	Unity matrix with size nxn

Symbols – Chapter 4

Symbol	Description
u_{fi}	"Failed" input signal of actuator i
σ_A	Failure parameter for actuator side
$\bar{u}$	Actuator bias
$\psi(t)$	Plant output
$\bar{y}$	Sensor bias
σ_S	Failure parameter for sensor side
$y(t)$	Sensor output
z	Augmented state
$f_z(z,u,v)$	Augmented system equations
$\bar{G}_v(t)$	Augmented noise input gain matrix
$F_z(t)$	Linearized system matrix of nonlinear model function
$H_z(\theta)$	Linearized observation matrix of nonlinear model function
$C_x(t)$	Linearized observation matrix of nonaugmented state space
$C_\theta(t)$	Linearized observation matrix of parameter part of state space
$\bar{R}_v(t), \bar{R}_r(t), \bar{R}_{vr}(t)$	Covariance matrices of augmented state space
$Q_z(\theta)$	Residual covariance matrix
$L_z(t)$	Augmented Kalman update gain
Q_p	Pseudo-noise mean-square intensity

Abbreviations

UAV	Unmanned Aerial Vehicle
MRAC	Model Reference Adaptive Control
ASG	Auxiliary Signal Generator
SISO	Single Input Single Output
MIMO	Multi-Input Multi-Output
HFG	High-Frequency Gain
MMAE	Multiple Model Adaptive Estimation
EMMAE	Extended Multiple Model Adaptive Estimation
EKF	Extended Kalman Filter
E{}	Expectation
$x(t+1\|t)$	x at time t+1 based on data at time t
$\hat{x}$	Estimation of x
$A^{(i)}$	Ith column of A
${}^{(0,i)}A$	Matrix A with ith column set to zero
${}^{(i,0)}A$	Matrix A with ith row set to zero

References

[BUT-92] Butler, H.: *Model Reference Adaptive Control - From Theory to Practice*, Prentice Hall International, 1992.

[EID-96] Eide, P., Maybeck, P.: *An MMAE Failure Detection System for the F-16*, IEEE Transactions on Aerospace and Electronic Systems Vol. 32, No. 3, July 1996.

[GEE-99] Geering, Hans P.: *Robuste Regelung*. Vorlesungsskript, Institut für Mess- und Regeltechnik, ETH Zürich, 1999

[GEE-03] Geering, Hans P.: *Regelungstechnik*. 6. Aufl. Springer-Verlag, Berlin, Heidelberg, 2003.

[GOO-84] Goodwin, G.C., and Sang Sin, K.: *Adaptive Filtering Prediction and Control*, Prentice Hall, Englewood Cliffs, NJ, 1984.

[GSX-04] Gang, T.; Shuhao, C.; Xidong, T.; Joshi S.M.: *Adaptive Control of Systems with Actuator Failures.* Springer-Verlag, London Berlin Heidelberg, 2004

[HIB-91] Hill, B.K., Walker, B.K.: *Approximate Effect of Parameter Pseudonoise Intensity on Rate of Convergence for EKF Parameter Estimators*, Proceedings of the 30th Conference on Decision and Control, Brighton, December 1991.

[HIL-03] Hiller, M.: *Computergestützte Kinematik und Dynamik von Mechanismen* . Vorlesungsskript, Institut für Mechatronik und Systemdynamik, Universität Duisburg-Essen, 2003

[JAZ-70] Jazwinski, A.H.: *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970

[LJU-79] Ljung, L.: *Asymptotic Behaviour of the Extended Kalman Filter as a Parameter Estimator for Linear Systems.* IEEE Transactions on Automatic Control, Vol. AC-24, No. 1, 1979, pp. 36-50.

[LJU-99] Ljung, L.: *System Identification: Theory for the User.* Prentice Hall, Englewood Cliffs, NJ, 1999.

[NAR-89] Narendra, K.S.; Annaswamy, A.M.: *Stable Adaptive Systems*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[SHA-96b] Shafai, E.: *Adaptive Regelung.* Vorlesungsskript, Institut für Mess- und Regeltechnik, ETH Zürich, 1996

[WAL-95] Walker, B.K., Huang, K-Y.: *FDI by Extended Kalman Filter - Parameter Estimation for an industrial Actuator Benchmark*, Control Engineering Practice Vol. 3, No. 12, 1995.